

National Institute of Technology, Raipur

Department of Computer Science and Engineering



Lab Report

Computer Network Lab

Submitted by:	Pratham Gandhi
Faculty name:	Prof. K. Jairam Naik
Semester:	Fourth
Roll Number:	20115903

Index

S. no.	Date of experiment	Name of experiment	Pg. no.	Date of submission	Remarks
1.	01-04-2021	Introduction to Local Area Network with its cables, connectors and topologies.	1	21-04-2021	
2.	01-04-2021	Installation of Switch. Hub their cascading and network mapping.	8	21-04-2021	
3.	05-04-2021	Installation of UTP, Co-axial cable, Cross cable, parallel cable NIC and LAN card.	10	21-04-2021	
4.	05-04-2021	Case Study of Ethernet (10 base 5,10 base 2,10 base T)	14	21-04-2021	
5.	06-04-2021	Installation and working of Net meeting and Remote Desktop.	19	21-04-2021	
6.	06-04-2021	Installation and working with Telnet (Terminal Network).	22	21-04-2021	
7.	07-04-2021	Installation and working with FTP (File Transfer Protocol).	24	21-04-2021	
8.	07-04-2021	Installation and Computers via serial or Parallel ports and enable the computers to share disk and printer port.	29	21-04-2021	
9.	08-04-2021	Installation of NS-2/3 Network Simulator: Basics of Network Simulation	31	21-04-2021	
10.	08-04-2021	Simulating a Local Area Network and LAN topologies.	34	21-04-2021	
11.	09-04-2021	Implementation of various MAC protocols.	37	21-04-2021	
12.	09-04-2021	Measuring Network Performance: Network Performance Evaluation, Performance Evaluation Metrics.	40	21-04-2021	
13.	12-04-2021	Performance Evaluation of routing Protocol.	42	21-04-2021	
14.	12-04-2021	Parameter Affecting the Performance of Network, Performance Evaluation Technique, Network Performance Evaluation using NS-2/3	44	21-04-2021	
15.	13-04-2021	Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.	46	21-04-2021	

S. no.	Date of experiment	Name of experiment	Pg. no.	Date of submission	Remarks
16.	13-04-2021	Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.	48	21-04-2021	
17.	14-04-2021	Write a program for error detecting code using CRC (16- bits)	51	21-04-2021	
18.	14-04-2021	Write a program to find the shortest path between vertices using bellman-ford algorithm.	54	21-04-2021	
19.	14-04-2021	Write a program for simple RSA algorithm to encrypt and decrypt the data.	57	21-04-2021	
20.	14-04-2021	Write a program for congestion control using leaky bucket algorithm.	59	21-04-2021	

EXPERIMENT-1

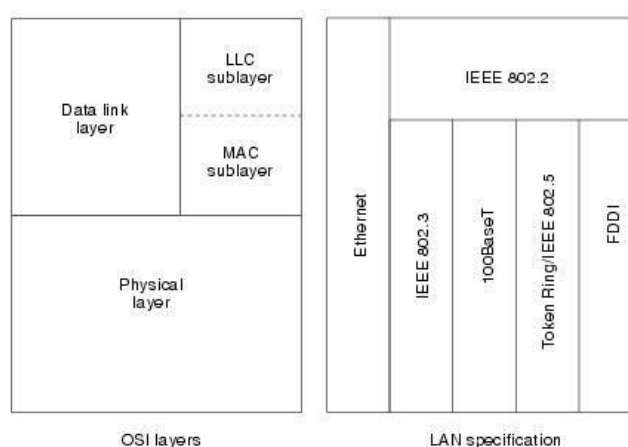
Aim: Introduction to Local Area Network with its cables, connectors and topologies.

LAN

Local Area Network (LAN) is a collection of devices that re connected together within a limited area in one physical location; for example, house, school, building etc. A LAN may thus range from having a single user in a personal-home network or thousands of users in an office.

LAN Protocols:

LAN protocols work at the two lowest layers of the OSI reference model, as shown in the figure below:



MAC Address:

MAC stands for **Media Access Control**, which is used in a LAN to identify the network devices. They are 48 bits long and contain 12 hexadecimal digits, first six of which and administered by the IEEE and they identify the manufacturer as they consist of a OUI (Organizational Unique Identifier), the latter six digits contain the serial number, and other values. These addresses are burned in ROM and thus are also called Burned-IN Addresses (BIA), they are later copied into the RAM during initialization. These addresses are supported in the data link layer of the OSI. According to the IEEE, the second layer has two components; the MAC Layer and the Logical Link Control Layer.

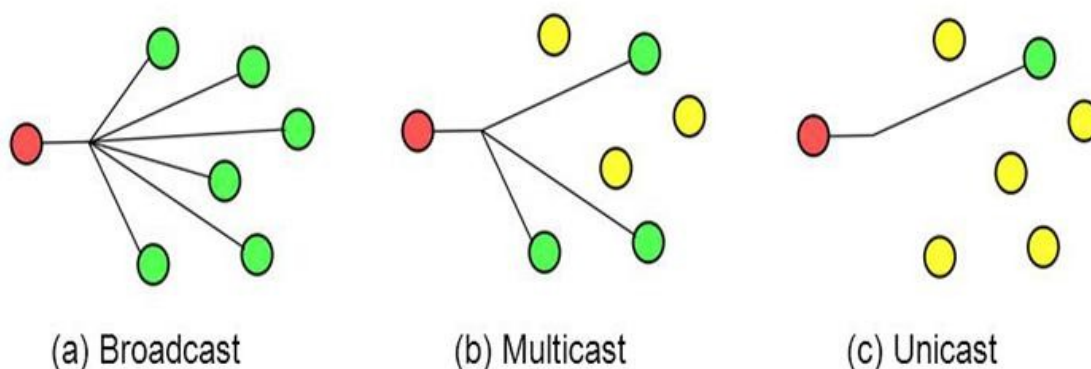
Network Layer Addresses:

These identify the device at the network layer (third layer) of the OSI. They are also called as virtual or logical address and exist within a hierarchical address space. It has two parts; the network (of which the device is a part) and number of devices on that network which means that the devices on same network have the same network part address.

LAN transmissions are either of the following three categories:

- 1) *Unicast*: A single packet is sent from source to the destination on a network (uni=>one). The packet is addressed by using the address of the destination node, after which it is transferred to the destination network which transfers it to the final destination.
- 2) *Multicast*: A single data packet is copied and forwarded to a predefined set of nodes on the network which are in the multicast address.
- 3) *Broadcast*: Broadcast transfers the network packet to all the nodes of the network and can be useful in sending a common notification or message.

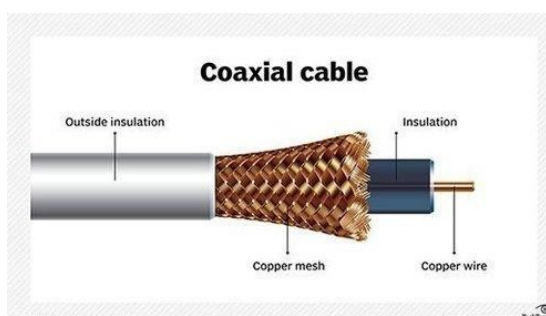
The following figure shows the depiction of the three types:



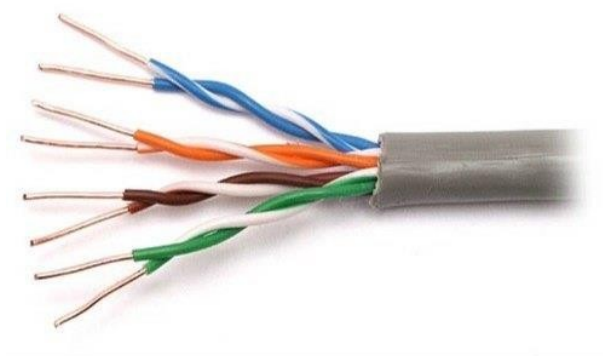
LAN Cables:

The LAN makes use of the different cabling for connecting the various nodes of the network. The three most commonly used cables in LAN are: Coaxial, Twisted pair and Fibre optic cables.

- 1) *Coaxial cabling*: In a coaxial cable there is an inner conductor that runs down through the middle of the cable and is called the core. This conductor is surrounded by a layer of insulation which is surrounded by a copper conducting mesh. All of which is surrounded by an insulation called as the sheath. This setup prevents any outside interference. It can be of two types: thinnet and thicknet. The maximum transmission speed is 10 Mbps. However, they are not used as much nowadays and are replaced by the twisted pair cables.

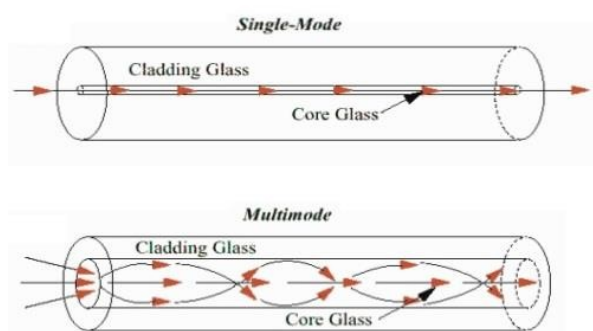


- 2) *Twisted pair cables*: This cable is primarily developed for computer networks and is also called as the ethernet cable. It is the most widely used cable by the modern day LANs. It consists of colour coded pair of twisted copper wires. There are usually four pairs of insulated copper and they are of two types: UTP (Unshielded Twisted pair), in which all the pairs are wrapped into a single plastic sheath and STP (Shielded Twisted Pair) in which each pair is additionally shielded by a metal shield and then they are wrapped into a single plastic sheath.



- 3) *Fiber Optic cable*: This cable consists of core, the thin strand carrying data over long distances and made up of glass or plastic; the cladding, it wraps the core and reflects the light back to the core; the buffer, which wraps the cladding and protects against any leakage of the light and jacket, which wraps buffer and protects the cable from any physical damage.

The Fibre Optic cable are the fastest cables that use the light to send data over forty kilometres at a speed of 100 Gbps. They are however quite costly and thus are not in use everywhere. Based on the beams of light transmitted at a given time, they can be classified as SMP (Single Mode Fibre) and MMF (Multi Mode Fibre).



LAN Connectors:

LAN makes use of different connectors for connecting the networks. The most common connectors are:

- 1) *USB*: It stands for the Universal Serial Bus and is used to eliminate the guesswork in connecting peripherals to a PC. A single port can be used for connecting up to 127 peripheral devices (such as mouse, speaker, keyboard etc.). It is expected to replace the serial and parallel ports.



- 2) *RJ-11*: Registered Jack-11 has four wires (RJ-12 has six), used for connecting telephone equipment. The following table lists the signals of the four wires:

RJ-11 Pin	Signal Name
1	VCC
2	Power Ground
3	One Wire data
4	One Wire Ground



- 3) *RJ-45*: Registered Jack-45 is an eight wires connector commonly used for connection of computers to a LAN. Even though they are larger than RJ-11 they can still be used for telephone equipment.



- 4) *MT-RJ*: Mechanical Transfer Registered Jack are often used with both single and multimode optical cables and are constructed with a plastic housing and provide an accurate alignment via metal guide pins and plastic ferrules.



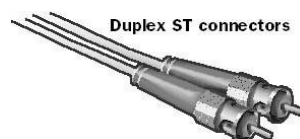
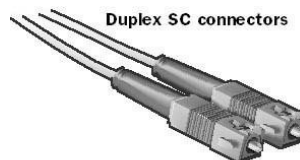
- 5) *Fibre LC*: Local Connectors are used for single and multimode fibre cables. They offer precise positioning with respect to transmitter's optical source emitter and receiver's optical detector.



- 6) *F-Type*: It is a kind of Rf connector used mainly for satellite television. It is inexpensive and offers good performance. The reason of its low cost is that it uses the centre wire of the coaxial cable as a pin of the male connector (which are cheaper than female connectors).



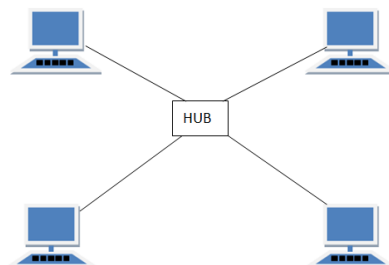
- 7) *ST and SC*: Both of these connectors are based on the concept that network requires two fibre cables; one for transmission and other for receiving. Each end is fitted with a plug that can be connected to a hub or switch. In North America Sc's are used (Subscriber Connector/Standard Connector) while Europeans use round ST's (straight Tips).



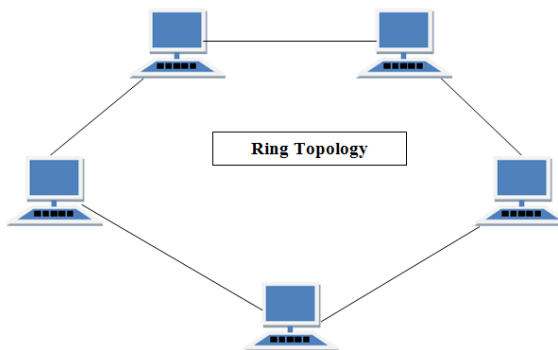
LAN Topologies:

Topology refers to how the devices are connected to each other; in LAN there are four major topologies:

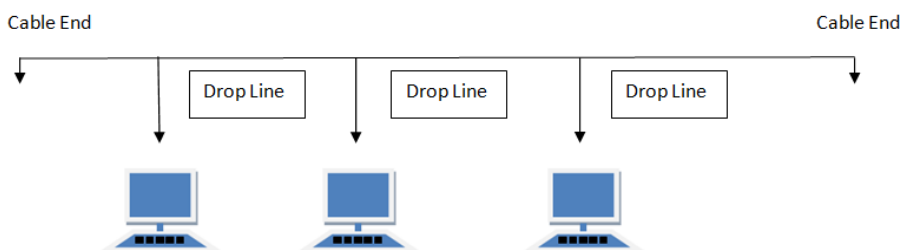
- 1) *Star Topology*: In this topology all the nodes are connected to a central point by cables like Shielded twisted Pair (STP), Unshielded twisted Pair (UTP), etc.



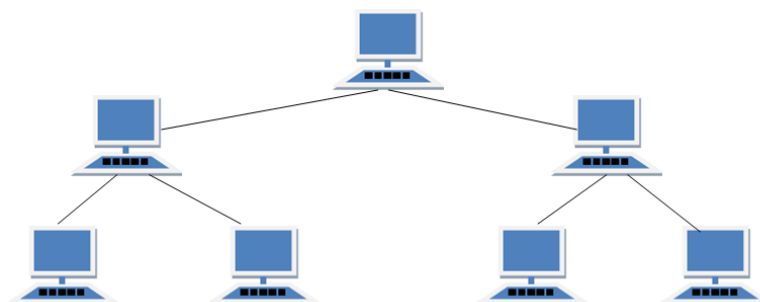
- 2) *Ring Topology*: All stations form a ring or a loop and consist of repeaters. There is no start or end of this networking.



- 3) *Bus Topology*: It makes use of a single length cable attached to all the LAN stations, which is its biggest weakness that if this single cable is broken the whole system fails.



- 4) *Tree topology*: It is an extension of the bus topology in a logical manner. There is a hierarchical structure and the network begins at the root node and there is only one active path between any two nodes and so there can be no loops.



Thus, we learnt about LAN through the various cables, connectors and topologies it uses and can conclude the following points:

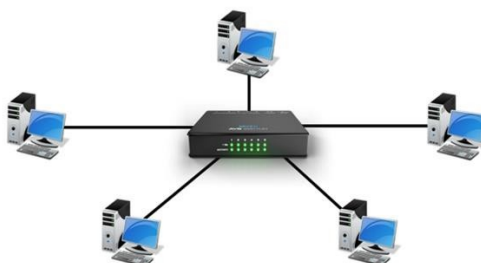
- LAN allows sharing of resources and data.
 - It helps establish client server relationships.
 - It provides easy and a fast way of communication.
 - It provides data security.
 - Even though it can only be used for a limited area, it provides a strong connection within the area.
-

EXPERIMENT-2

Aim: Installation of Switch. Hub their cascading and network mapping.

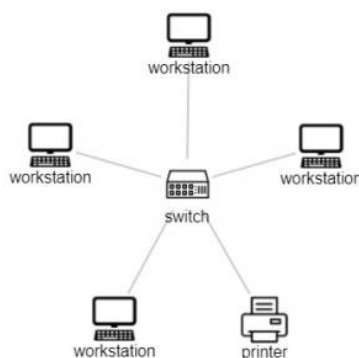
Hub:

- A hub is a multiport repeater. It connects multiple wires that come from different branches; for example, in topologies.
- It is a physical layer networking device used to connect multiple devices in network (such as LAN), the computer that needs to be connected should be plugged in into one of its ports.
- Hub cannot take an intelligent decision so when a data pack arrives it transports it to all the devices connected through its ports, without considering the correct destination or the better route leading to inefficiency and wastage.



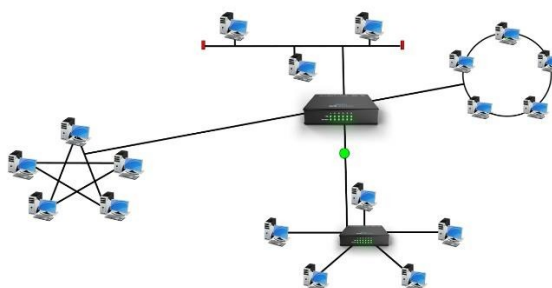
Switch:

- A switch is a multiport device which also consists of a buffer and a design to boost the efficiency and performance.
- It is a data link layer device which can check for errors before forwarding any data and hence forwards only selective correct data packets to the destined port only.
- It can be used to connect many devices to a network through its port and when a data packet arrives it sends it to the corresponding devices by reading the destination address and hence supports all unicast, multicast and broadcast communication.



Router:

- Router is like a switch that routes the data packets based on their IP addresses and is a Network Layer device.
- It is normally used for connecting the LANs and WANs together.
- It consists of a dynamically updating routing table based on which it makes a decision on routing the data packets.

**Differences between Hub and Switch:**

KEY	HUB	SWITCH
Objective	To transmit the signal to port to respond to where the signal was received.	To provide connection and termination settings based on need.
Layer	They operate in the physical layer of the OSI reference model.	They operate in the data link layer of the OSI reference model.
Transmission	It only broadcasts data packets. (sends it to all the receivers).	It can unicast, multicast or broadcast a data packet.
Feature	It is a non-intelligent device.	It is an intelligent device.
Active/passive device	It is a passive device as there is no software attached to it.	It is an active device with a software attached to it.
Transmission Mode	Transmission mode is half duplex.	Transmission mode if full duplex.
Number of ports	They have fewer ports generally (maximum ports being 4).	They have higher number of ports (24-28 ports).

Procedure:

- 1) Connect all the devices via LAN cable to the Hub/Switch.
- 2) Assign IP addresses by using the command *IP Config*.
- 3) Check connectivity for making sure they are correctly connected by using the command *Ping ip_address* (where write down the IP address of the required device).

EXPERIMENT-3

Aim: Installation of UTP, Co-axial cable, Cross cable, parallel cable NIC and LAN card.

UTP

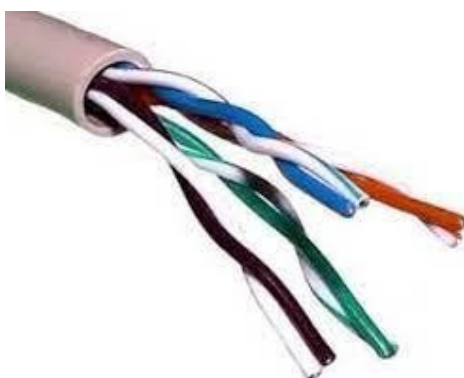
- UTP stands for Unshielded Twisted Pair cable. It is basically a 100 ohm copper cable that consists of 2 to 1800 unshielded twisted pairs surrounded by an outer jacket.
- It is called as unshielded as it has no metallic shield.
- It is small in diameter (because of the lack of shield) but is also unprotected against electrical interference.
- The twisted cables improve its immunity to electrical noise and EMI.

Categories of UTP:

Category	Speed	Use
1	1 Mbps	Voice only
2	4 Mbps	LocalTalk and Telephone
3	16 Mbps	10BaseT Ethernet
4	20 Mbps	Token Ring
5	100 Mbps (2 pair)	100BaseT Ethernet
	1000 Mbps (4 pair)	Gigabit Ethernet
5e	1,000 Mbps	Gigabit ethernet
6	10,000 Mbps	Gigabit Ethernet

UTP cable connectors: RJ-45

- RJ-45 (Registered Jack-45) is an eight wire connector used to connect computers to a LAN and mostly with Ethernet cables.
- It is a standard connector which features eight pins to which the wire strands of cable interface electrically.
- RJ-11 closely resemble it and is slightly narrower, used for telephone cables.



Coaxial Cable:

They are commonly called as coax, are copper cables with metal shielding that provide immunity against noise and a greater bandwidth. It can transmit signals over a large distance at a higher speed if compared to the twisted pair cable.

Structure of coaxial cable:

It has a central core of copper which transmits the signals. It is covered by an insulating material. This insulator is again woven into a outer metal conductor that acts as a shield for noise, which is then coated with a insulating cover.

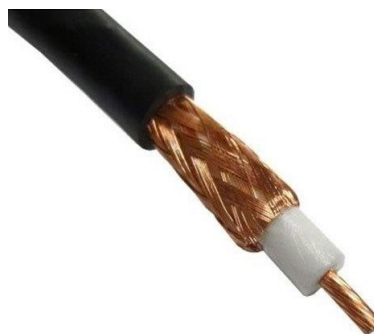
Categories of coaxial cables:

The cables are categorixed into three categoriesas per radio Government (RG) readings:

- 1) RG-59; It has impedance of 75 W and used in cable TV.
- 2) RG-58; It has impedance of 50 W and used in thin Ethernet cables.
- 3) RG-11: It has impedance of 50 W and used in thick ethernet cables.

Applications of coaxial cables:

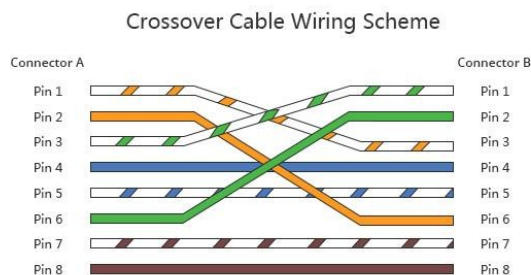
- They are used in analog telephone networks and can carry upto 10,000 voice signals.
- They can be used in digital telephone networks with a speed upto 600 Mbps.
- These cables can be used in traditional Ethernet LANs and MANs.
- They can be used in cable TV networks.



Ethernet cross-over cables:

- It is used for connecting two Ethernet network devices directly without a switch or router in between by connecting the transmitting pins of one side by receiving pins of other side.
- They can send and receive data by enabling complex data transfers,
- They have a crisscross pair of wires; which reverses the incoming and outgoing signals.
- It can be used for connecting:
 - Two computers
 - Two hubs
 - A hub to a switch
 - A cable modem and a router

■ Two router interfaces



NIC:

NIC stands for Network Interface Card is a hardware component that is a prerequisite for the computer to be connected to the internet. It is basically a circuit board installed in a computer that provides dedicated network connection to the computer. It is also called as Network Interface controller/Network Adapter/LAN Adapter/LAN Card.

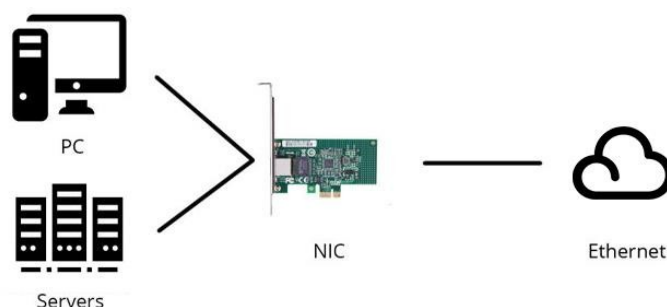
It is of two types:

Internal network cards: They are inserted in a predefined slot of motherboard and requires network cables.

External network cards: If there is no internal NIC in desktops or laptops, then it is used and can be wither wireless or USB type.

Purpose of NIC:

- It allows both wired and wireless communication.
- It is in both physical and data link layer.
- It allows communications between computer connected through LANs as well as communications over large-scale network through Internet Protocol (IP).



Procedure:

The following steps discuss the procedure to make your own Ethernet cable:

Materials Required:

- Unshielded Twisted Pair (UTP) Patch cable
- Modular connector (8P8C plug, aka RJ45)
- Crimping tool
- Cable tester (optimal, but recommended)

Steps:

- 1) Strip the jacket of cable up to about 1.5 inches down from the end.
 - 2) Spread the four pairs of twisted wire apart, for category 5e pull string can be used.
 - 3) Untwist the wire pairs and neatly align them in the T568B orientation.
 - 4) Cut the wires now (as straight as possible) about 0.5 inches above the end of the jacket.
 - 5) Insert the wires all the way into the modular connector carefully and make sure that each wire passes through the appropriate guides inside the connector.
 - 6) Push the connector inside the crimping tool and squeeze it all the way down.
 - 7) Repeat the steps 1-6 for the other end of the cable too.
 - 8) Use a cable tester for each pin to make sure that you have terminated each end of the cable successfully.
-

EXPERIMENT-4

Aim: Case Study of Ethernet (10 base 5,10 base 2,10 base T).

Ethernet:

It is a frame-based computer networking technique for local area networks (LANs). The name 'Ethernet' is derived from the word *ether* and its physical concepts. It is the most widely used method of linking computers since the 1990's. The basic idea behind its design is that multiple computers have access to it and can send data any time. It is standardized as IEEE 802.3.

There are various variants of Ethernet, we are going to look into three of these variants: 10BASE2, 10BASE5 and 10BASE-T.

10BASE2:

It is a variant of Ethernet that uses thin coaxial cable (RG-58 or similar) and thus is called as cheapernet, thin ethernet, thinnet, thinwire. It is terminated with BNC connectors. It was a dominant 10 Mbit/s standard for years when it came into existence, however it has now become obsolete with popularity of 802.11 wireless networks.

Network Design:

In this cable each segment of the cable is connected to the transceiver using a BNC-T connector. At the physical end of the network, a terminator is required of 50 ohms. This is most commonly connected directly to the T-connector on a workstation though it does not technically have to be.

Wiring should be done carefully with cables properly connected and the appropriate terminators installed. One terminator is connected to ground through a ground wire. A failure at any point prevents any communication and bad contacts and shorts and very difficult to diagnose. And thus these networks are very difficult to maintain, and so an upgrade (10BASE-T) was required.

Characteristics:

- They cannot be extended without breaking services temporarily for existing users and the presence of many joints in the cable makes them vulnerable to accidents, or malicious disruptions.
- They do have some advantages like: The low hardware cost, and thus is ideal for smaller networks involving two or more computers.



10BASE5

These variants are also called as the thicknet as it a thick and stiff 0.375 inch diameter and 500 metres in length cable with an impedance of 50 ohms. It has a solid centre conductor, a foam insulating fiber, a shielding braid and an outer jacket. Since the outer sheath is yellow-to-orange in colour, they are also called as the yellow cable or the orange cable. It is obsolete in today's time but can be found in the older systems where it was used earlier.

Network Design

They are designed to allow transceiver to be added while existing connections are live. It is achieved by vampire tap-a device which (with sufficient practice) clamps onto the cable, forcing a spike to pierce through the outer shielding to contact the inner conductor while other spikes bite into the outer conductor. The transceiver can also be connected by using N connectors at the end of cable segment.

The maximum number of nodes that can be connected to this cable are limited to 100 and transceivers can be installed only at 2.5 metre intervals. This distance is so chosen to avoid the reflections from multiple taps not in phase, and these points are marked on the cable with black bands. A 50-ohm resistive terminator is required at each end of the cable.

The transceiver is connected to the node using an interface called the Attachment Unit Interface (AUI), which has 15 pins, two-row d-style connector but with clips instead of the more normal screws for cable restraint.

Characteristics:

- These systems are difficult to install and maintain.
- 10BASE5, is derived from 10Mbit/s (its speed), BASE (which is short for baseband signaling) and 5 stands for maximum segment length of 500 metres.
- With termination missing, or if there is a break in the cable, the signal on the bus will be reflected, rather than dissipated when it reaches the end, this reflect signal is indistinguishable from a collision and prevents any kind of communication.



10BASE-T

They are called as the Ethernet over twisted pair or the copper based computer networking physical connectivity methods. These are the most widely used cables, their subtypes: 10BASE-T, 10BASE-TX, 1000BASE-T run at 10Mbit/s, 100Mbit/s, 1000Mbit/s respectively. All of these three standards use the same connectors. Since these are high speed cables they can also be mixed with the older generations as well. They use 8 position modular connectors. The cables usually used are four-pair Category 5 or above twisted pair cable. Each of the three standards support both full duplex and half-duplex communication. According to the standards, they all operate over distances of up to 100 meters.

Its name is derived from several aspects of the physical media. The number refers to the theoretical maximum transmission speed in Megabits per second (Mbit/s). The BASE is short for baseband. The T designates twisted pair cable, where the pairs of wires are twisted together for purposes of reducing crosstalk.

Characteristics:

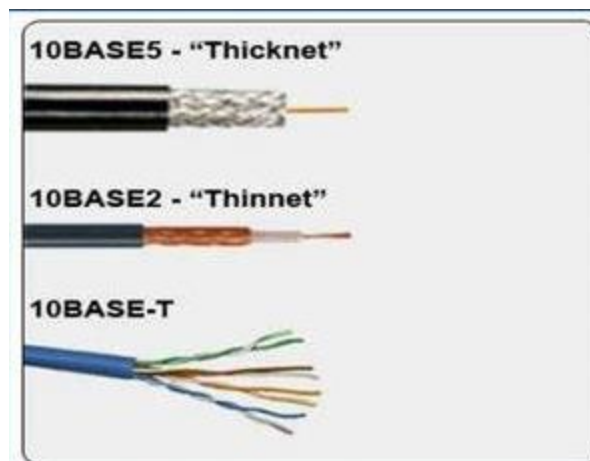
- Twisted-pair Ethernet standards are such that the majority of cables can be wired 'straight-through' (pin 1 to pin 1, pin 2 to pin 2 and so on), but others may need to be wired in the 'crossover' form (receive to transmit and transmit to receive).
- 10BASE-T and 100BASE-TX only require two pairs to operate, pins 1 and 2 (transmit or TX), and pins 3 and 6 (receive or RX). Since 10BASE-T and 100BASE-TX need only two pairs and Category 5 cable has four pairs.
- This is not possible with 1000BASE-T since it requires all four pairs to operate, pins 1 and 2, 3 and 6 — as well as 4 and 5, 7 and 8. It is conventional to wire cables for 10 or 100 Mbit/s Ethernet to either the T568A or T568B standards.
- A 10BASE-T node (such as a PC) that transmits on pins 1/2 and receives on pins 3/6 to a network device is most often on a "straight-through" cable in the "MDI" wiring pattern where RX goes to RX and TX goes to TX

- Many modern Ethernet host adapters can automatically detect another PC connected with a straight-through cable and then automatically introduce the required crossover, if needed; if one or neither of the PC does not, then a crossover cable is required. If both devices being connected support 1000BASE-T according to the standards, they will connect regardless of the cable being used or how it is wired.
- To connect two hubs or switches directly together, a crossover cable can be used, but some hubs and switches have an “uplink” port used to connect network devices together, or have a way to manually select MDI or MDI-X on a single port so that a straight-through cable can connect that port to another switch or hub.
- 100BASE-TX follows the same wiring patterns as 10BASE-T but is more sensitive to wire quality and length, due to the higher bit rates.
- 1000BASE-T uses all four pairs bi-directionally and the standard includes auto MDI-X, however implementation is optional. With the way that 1000BASE-T implements signaling, how the cable is wired is immaterial in actual usage. The standard on copper twisted pair is IEEE 802.3ab for Cat 5e UTP, or 4D-PAM5; 4 Directions using PAM (pulse amplitude modulation) with 5 voltages, -2, -1, 0, +1, and +2
- Unlike earlier Ethernet standards using broadband and coaxial cable, such as 10BASE5 (thicknet) and 10BASE2 (thinnet), 10BASE-T does not specify the exact type of wiring to be used but instead specifies certain "characteristics" which a cable must meet.
- 100BASE-TX and 1000BASE-T both require a minimum of Category 5 cable (5e or 6 with 1000) and also specify a maximum cable length of 100 meters. Furthermore while 10BASE-T is more tolerant of poor wiring such as split pairs, poor terminations and even use of short sections of flat cable, 100BASE-T is not as much so, and 1000BASE-T is less tolerant still.
- Since testing of cable is often limited to checking if it works with Ethernet, running faster speeds over existing cable is often problematic. This problem is made worse by the fact that Ethernet's auto negotiation takes account only of the capabilities of the end equipment not of the cable in between.



TIA/EIA-568-A T568A Wiring			
Pin	Pair	Wire	Colour
1	3	tip	white/green
2	3	ring	green
3	2	tip	white/orange
4	1	ring	blue
5	1	tip	white/blue
6	2	ring	orange
7	4	tip	white/brown
8	4	ring	brown

10BASE-2, 10BASE-5 and 10BASE-T are all variants of ethernet. However, the former two are not used anymore. They have similarities to the ethernets but differ in the characteristics mentioned above. The picture below shows the fundamental difference between them:



EXPERIMENT-5

Aim: Installation and working of NetMeeting and Remote Desktop.

NetMeeting:

- Microsoft Windows NetMeeting is used for real time PC-to-PC screen sharing and remote control of Windows PC's across the network.
- It can be used in parallel with the main video conferencing system so that any site connected to it can share the screen or application of other sites.
- It was included with Microsoft Windows versions 95 OSR2 to Windows XP.

Installation of NetMeeting:

- 1) Download the NetMeeting software from Microsoft free of cost.
- 2) Install it, while installing it asks whether we would want to log on to a directory server when NetMeeting starts.
- 3) If we select this option others will be able to see our name (which might result in unsolicited callers).
- 4) Once the program is installed, launch it and finish the configuration.

Activating remote desktop sharing:

- 1) Launch NewMeeting.
- 2) Choose **Tools-> Enable Sharing** and follow the instructions.
- 3) Reboot the computer.
- 4) After rebooting, launch NewMeeting.
- 5) Choose **Tools->Remote Desktop Sharing** and follow instructions on screen.
- 6) Close NetMeeting.

Running a remote session:

- 1) Launch NewMeeting.
- 2) Click on **Place Call** button.
- 3) On the top bar types the IP address or name of the remote computer.
- 4) If **Require Security for this call** check box isn't selected, select it to turn on security.

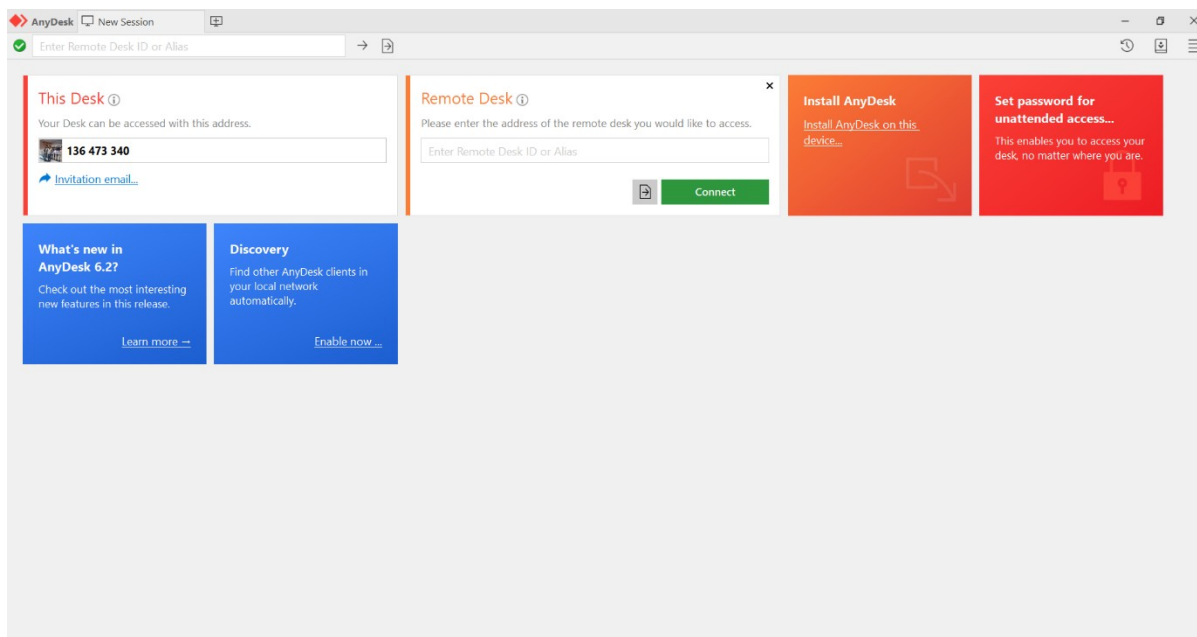


Anydesk:

It is a remote desktop software that can be run like a normal program. It ensures secure remote desktop connections for IT professionals.

Features:

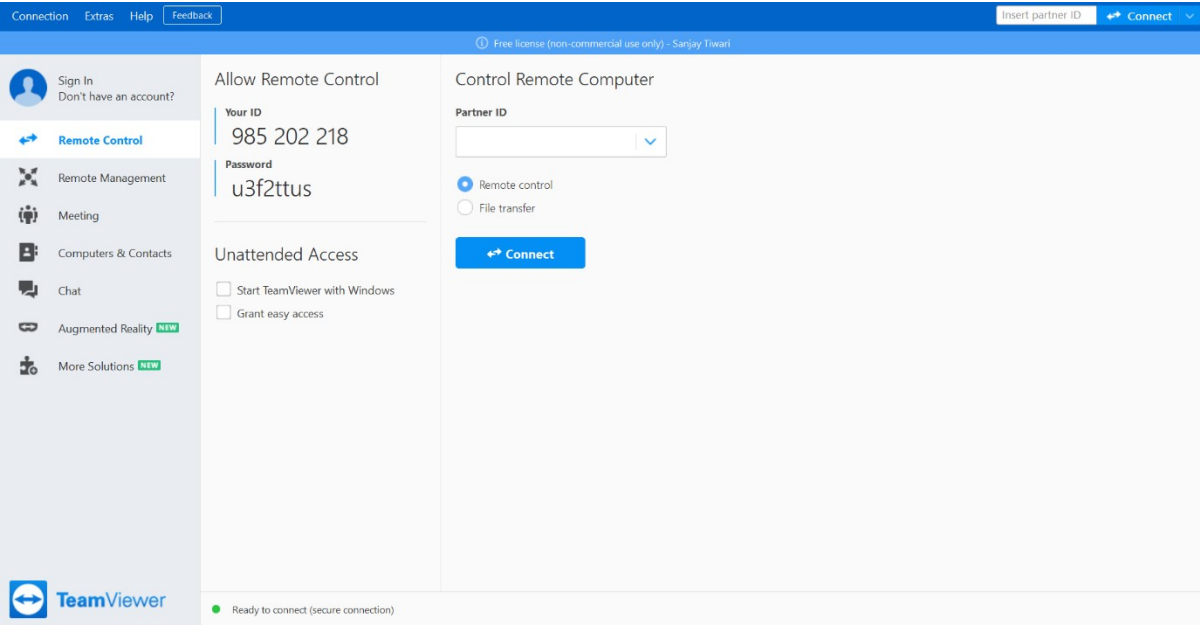
- Supports audio and file transfer.
- Connections can be configured to favor quality or speed.
- Works on Linux, MacOS, Windows.
- Feature of recording the remote session.
- Auto updates for new versions.

**TeamViewer:**

It is a remote desktop software allowing to connect workstations remotely. It enhances the remote control performance with GPU for hardware accelerated image processing.

Features:

- Allow you to pass the control from one person to other.
- It supports Augmented reality.
- Multiple parties can be included in connection.
- Files can be dragged and dropped from one computer to another.
- Easy and secure.



EXPERIMENT-6

Aim: Installation and working with Telnet (Terminal Network).

Telnet:

- It is a popular client-server program used to provide a general client-server environment where the user can access any application program of remote computer.
- It stands for Terminal Network.
- It provides connection to remote computer in a way that local terminal appears to be at remote side.

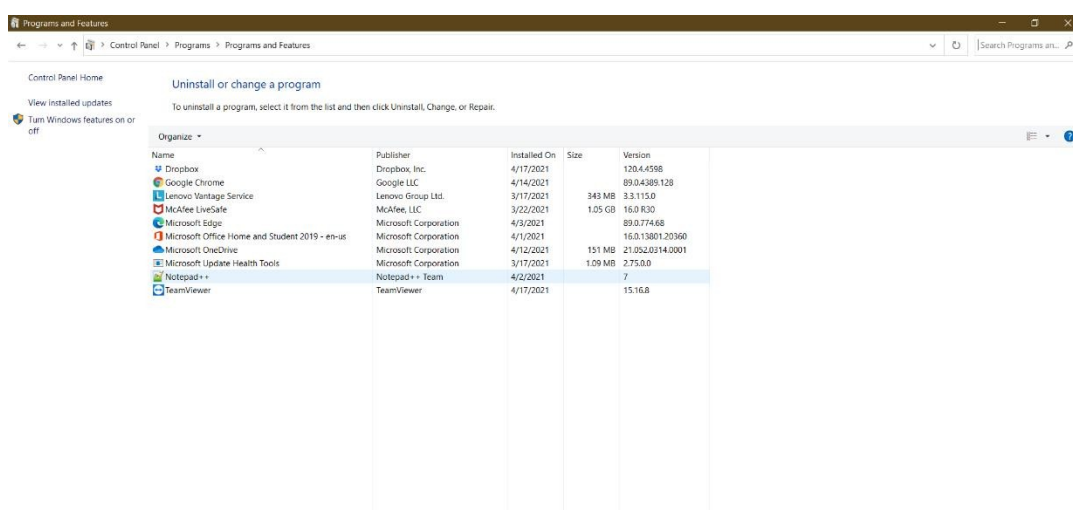
Installation of Telnet on Windows:

Unlike Linux Telnet is not preinstalled in on Windows, it can be installed by following steps:

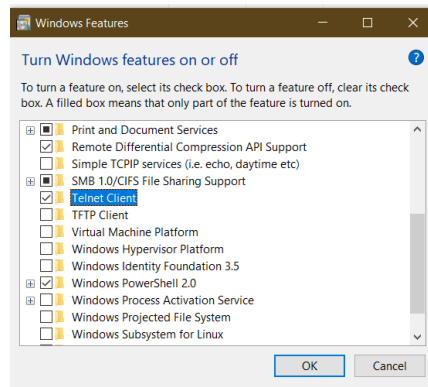
- 1) Click on **Start**.
- 2) Select **Control Panel**.



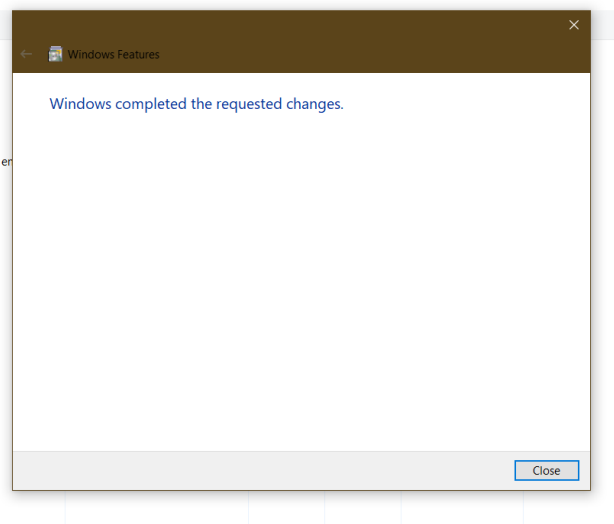
- 3) Choose **Program and Features**.



- 4) Click **Turn Windows features on or off**.
- 5) Select the **Telnet Client** option.



- 6) Click OK.
- 7) A dialogue box appears to configure the installation. The Telnet commands are then available.



EXPERIMENT-7

Aim: Installation and working with FTP (File Transfer Protocol).

FTP:

File Transfer protocol server can be set up on Windows to upload and download files from virtually anywhere to the computer without the limitations of the cloud storage. It creates a private cloud without any file or size restrictions and can be used for multiple accounts. Thus, it proves to be very useful personally and professionally.

Installation of FTP server components:

1. The required components of FTP can be added manually.
2. Open *Control Panel*.
3. Click on *Programs*.
4. Under *Programs and features* click the *Turn Windows features on or off*.
5. When the dialog box opens, expand the *Internet Information Services* feature and expand *FTP server* option.
6. Check the *FTP Extensibility* and *FTP Service* option.
7. Check the *Web Management Tools* option with the default selections, but making sure that the *IIS Management Console* option is checked.
8. Click the OK button.
9. Click the Close button.

Configure FTP server site on Windows 10

After installing the required components, configure an FTP server on the computer, which involves creating a new FTP site, setting up firewall rules, and allowing external connections.

Setting up an FTP site

- 1) Open *Control Panel*.
- 2) Click on *System and Security*.
- 3) Click on *Administrative Tools*.
- 4) Double-click the *Internet Information Services (IIS) Manager* shortcut.
- 5) On the *Connections* pane, right-click *Sites*, and select the *Add FTP Site* option.
- 6) In the FTP site name, type a short name for your server.
- 7) In the *Content Directory* section, under *Physical path* click the button on the right to locate the folder you want to use to store your FTP files.
- 8) Click the Next button.
- 9) Use the default *Binding settings* selections.
- 10) Check the *Start FTP site automatically* option.
- 11) In the "SSL" section, check the *No SSL* option.
- 12) Click the Next button.
- 13) In the *Authentication* section, check the *Basic* option.
- 14) In the *Authorization* section, use the drop-down menu, and select *Specified users* option.

- 15) Type the email address of your Windows 10 account or local account name to allow yourself access to the FTP server.
- 16) Check the *Read and Write* options.
- 17) Click on the *Finish* button.

FTP is now operational on the computer.

Configuring Firewall Rules:

The services should be manually allowed as the built-in firewall on Windows-10 will block the FTP server.

- 1) Open *Windows Defender Security Center*.
- 2) Click on *Firewall & network protection*.
- 3) Click the *Allow an app through firewall* option.
- 4) Click the *Change settings* button.
- 5) Check the *FTP Server* option, as well as the options to *allow Private and Public access*.

Allowing External Connections:

For reaching external servers, TCP/IP port number 21 has to set up.

- 1) Open *Settings*.
- 2) Click on *Network & Internet*.
- 3) Click on *Status*.
- 4) Click the *Change connections properties* option.
- 5) Make a note of the IPv4 DNS server address, which is the address of your router. Usually, it's private address in the 192.168.x.x range. For instance, 192.168.1.1 or 192.168.2.1.
- 6) Open your default web browser.
- 7) On the address bar enter the router's IP address and press Enter.
- 8) Sign-in with your router credentials.
- 9) Open the *Port Forwarding* page.
- 10) Add a new rule to forward incoming connections to the FTP server from the internet by including this information:
Service name: Type a descriptive name for the port forwarding rule.
Port range: 21.
Local IP: This is the FTP server IP address that the router will forward incoming connections.
Local port: 21.
Protocol: TCP
- 11) Click the add button.
- 12) Click the Apply button to save the changes.

After completing the steps, any incoming connection on port 21 will be forwarded to the FTP server to establish a networking session

Setting up static IP address:

- 1) Open *Control Panel*.

- 2) Click on *Network and Internet*.
- 3) Click on *Network and Sharing Center*.
- 4) In the left pane, click the *Change adapter settings* option.
- 5) Right-click the network adapter, and select the *Properties* option.
- 6) Select the *Internet Protocol Version 4 (TCP/IPv4)* option.
- 7) Click the *Properties* button.
- 8) Select the *Use the following IP address* option.
- 9) Specify the IP settings:
IP address: Specify a static network address for the computer.
Subnet mask: In a home network, the address usually is 255.255.255.0.
Default gateway: This is usually the IP address of the router.
Preferred DNS server: Typically, this is also the IP address of your router.
- 10) Click the OK button.
- 11) Click the Close button.

Set up multiple FTP accounts on Windows 10:

If you want to allow multiple people to download and upload files to the FTP server simultaneously, you need to set up multiple accounts with specific permissions

Creating new user accounts:

- 1) Open *Settings*.
- 2) Click on *Accounts*.
- 3) Click on *Family & other people*.
- 4) Click the *Add someone else to this PC* button.
- 5) Type the Microsoft account address for the user you want to allow access to the FTP server.

Configuring user accounts to FTP server:

- 1) Open *Control Panel*.
- 2) Click on *System and Security*.
- 3) Click on *Administrative Tools*.
- 4) Double-click the *Internet Information Services (IIS) Manager* shortcut.
- 5) On the left pane, expand *Sites*, and select the site you created earlier.
- 6) Double-click the *FTP Authorization Rules* option.
- 7) On the right pane, click the *Add Allow Rule* option.
- 8) Select one of these two options:
All Users: Allows every user configured on your Windows 10 device to access the FTP server.
Specified users: You can use this option to specify all the users you want to access the FTP server. (separate each user using a comma.)
- 9) Check the *Read and Write* options.
- 10) Click the OK button.

Connect to an FTP server remotely on Windows 10

Viewing and downloading files:

- 1) Open a web browser.
- 2) In the address bar, type the server IP address using ftp://, and press Enter. For example, ftp://192.168.1.100.
- 3) Type your account credentials.
- 4) Click the *Log in* button.

After these steps, you will be able to navigate and download files and folders from the server.

Viewing, downloading and uploading files:

- 1) Open File Explorer.
- 2) In the address bar, type the server address using ftp://, and press Enter. For example, ftp://192.168.1.100.
- 3) Type your account credentials.
- 4) Check the *Save password* option.
- 5) Click the *Log on* button.

After completing these steps, you can browse files and folders, and download and upload files as if they are locally available on your device.

Configuring multi-users:

There are multiple procedures for doing so to allow others to access the FTP server:

Create multiple FTP accounts:

- 1) Open settings app. (Windows key + I.)
- 2) Click *accounts*.
- 3) Navigate to the left-side pane, where you'll see an option called *family & other users*. Click the link.
- 4) On the right-hand side, look for the option *Add someone else to this PC*.
- 5) In the next screen, you'll be prompted to enter that person's email ID or phone number. Look for a link called *I don't have this person's sign in information* at the bottom left. Click the link.
- 6) In the next screen, you'll be asked to enter the personal details of the new user. If you know the other person's Microsoft account details, you can enter. Otherwise, look for a link called *add a user without a Microsoft account* on the bottom left.
- 7) In the next screen, enter the details of the user. Choose a strong password for your own security as this server can be accessed from the Internet.

Add a new user to the FTP folder

- 1) Navigate to the FTP folder, right-click on it and select *properties*.
- 2) Click the *security* tab and navigate to *edit* button.

- 3) Look for *Add* button in the next screen and click on it. This will open the *select users or groups* dialog box. In this screen, enter the name of the user you want to add in the text area. Click on *check names* button to make sure that you have entered the right name.
- 4) Click OK
- 5) Select the newly added user account and set the permissions for this user. Once done, click apply and OK.

Configure the user to access the FTP folder

- 1) Go to *control panel* and open *administrative tools*.
- 2) Double-click *Internet information services (IIS) manager*.
- 3) Expand the left-hand pane and navigate to sites. Click on sites and you'll see your FTP server. Select this server.
- 4) On the right-hand pane, you'll see many options. Look for *authorization rules*. Right-click on it and select *add allow rules* from the context menu.
- 5) In the next screen, choose *selected users* option and enter the Windows 10 account you created earlier. Set the permission you'd like this user to have. Click OK.

Windows has an in-built feature for setting up the FTP server, and so it can be set up with great ease. FTP provides lot of flexibility and convenience and even allow multiple users addition so that sending and receiving files could be made easier.

EXPERIMENT-8

Aim: Installation and Computers via serial or Parallel ports and enable the computers to share disk and printer port.

Port:

A port is a point at which an external device (hardware) can be attached to the computer system. There are two kinds of ports:

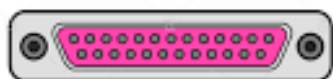
- 1) *Serial Port:* They transmit data sequentially one bit at a time. They are usually 9-pin or 25-pin male connectors.

Serial Port



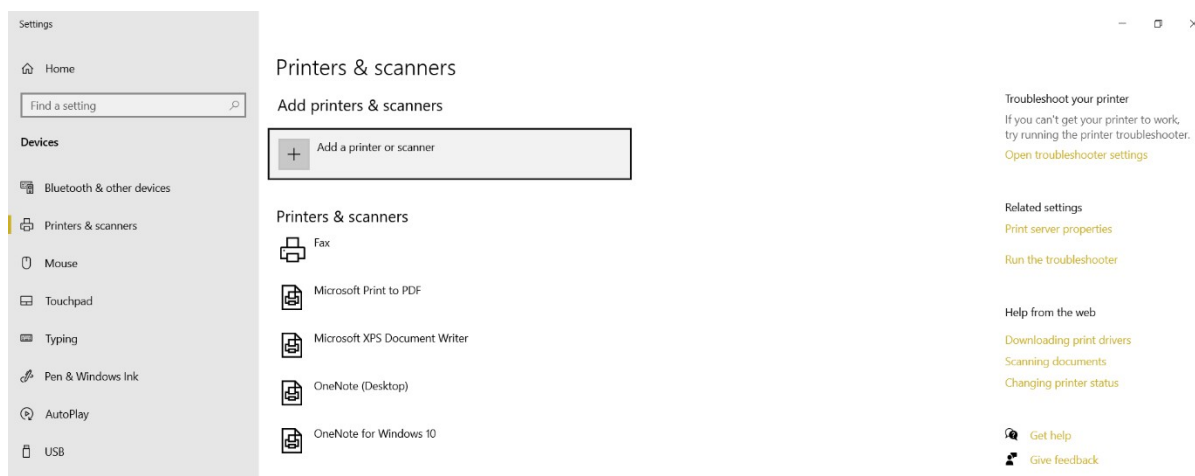
- 2) *Parallel Ports:* Parallel ports send or receive data at speed 8 bits (1 byte) at a time. They are 25-pin female pin are used for connecting printers, scanners, HDD etc

Parallel Port



Adding a Network Printer to Windows Computer:

- 1) Click on the **start** button, and then select **Devices and printers**.
- 2) In this panel. Click on **Add a printer**.



- 3) In the Add Printer window, click on the option **Add a local printer**.
- 4) Select **Create a new port** and then select **Standard TCP/IP Port** from the drop down menu. Click Next.
- 5) Enter the IP Address of the printer and select the option **Query the printer and automatically select the driver to use**. Click Next.

- 6) Select the option **Use the driver that is currently installed.**
- 7) Enter the descriptive name for the Printer.
- 8) Select on **Do not share this printer** and click Next.
- 9) Print a test page for confirmation.
- 10) Click close.
- 11) Click Finish.

You can check the name of the printer in the devices and printers list.

EXPERIMENT-9

Aim: Installation of NS-2/3 Network Simulator: Basics of Network Simulation.

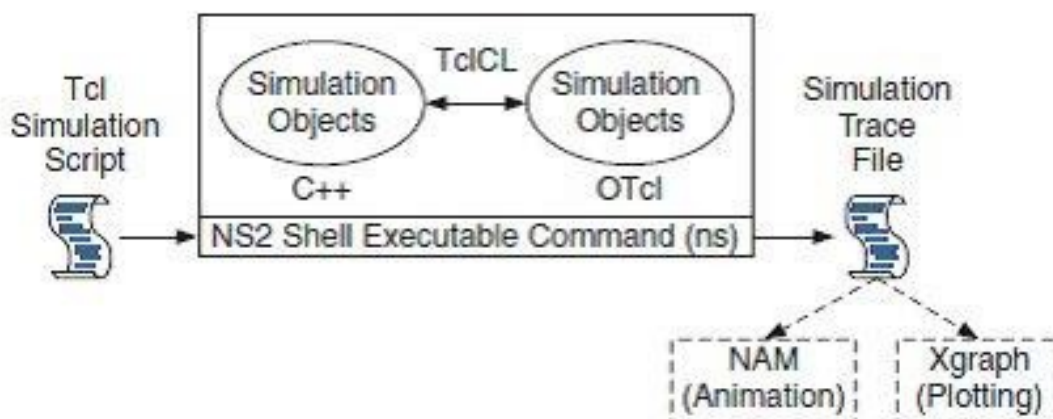
NS-2:

- It stands for Network Simulator-2. It is an event driven simulation tool that is widely used for studying the dynamic nature of the communication networks.
- It can be used for simulating wired as well as wireless network function and protocols like TCP/IP, UDP etc.
- It provides a method for the user to specify network protocols and simulate their corresponding behaviour.

Features of NS-2:

1. It simulates wired and wireless networks.
2. It is primarily Unix-based.
3. It is a discrete simulator.
4. It provides support to a variety of protocols: TCP, FTP, UDP, http and DSR.
5. Uses TCL as its scripting language.

Structure of NS-2:



Basic architecture of NS.

Installation of NS-2 on Ubuntu:

Step-1: Update and install prerequisite packages:

Using the commands:

- `$sudo apt-get update`

- \$sudo apt-get dist-upgrade
- \$sudo apt-get update
- \$sudo apt-get gcc
- \$sudo apt-get install build-essential autoconf automake
- \$sudo apt-get install tcl8.5-dev tk8.5-dev
- \$sudo apt-get install perl xgraph libxt-dev libx11-dev libxmu-dev

Step-2: Download ns-allinone-2.35.tar.gz.

Step-3: Select the file and click the right mouse button and choose option 'Extract Here', to extract its contents.

Step-4: After the extraction is complete, types the following command:

```
$cd ns-allinone-2.35
```

Step-5: Run the following command:

```
./install
```

Step-6: Set the path for NS-2 global access:

Command: \$sudo gedit ~/.bashrc

A file opens with extension .bashrc

Set path in that file:

```
# LD_LIBRARY_PATH
OTCL_LIB=/Yourpath/ns-allinone-2.35/otcl-1.14
NS2_LIB=Yourpath/ns-allinone-2.35/lib

X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR
_LOCAL_LIB
# TCL_LIBRARY
TCL_LIB=/Yourpath/ns-allinone-2.35/tcl8.5.10/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=/Yourpath/ns-allinone-2.35/bin:/Yourpath/ns-allinone-
2.35/tcl8.5.10/unix:/Yourpath/ns-allinone-2.35/tk8.5.10/unix
NS=/Yourpath/ns-allinone-2.35/ns-2.35/
NAM=/Yourpath/ns-allinone-2.35/nam-1.15/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

After setting the path, save the file and reboot your system, and type command: ns
The symbol % will appear in command prompt, which would conclude that the NS-2 is
correctly installed in your system.

EXPERIMENT-10

Aim: Simulating a Local Area Network and LAN topologies.

Instructions for execution:

- 1) We will write a tcl script and simulate it by ns2.
- 2) We begin by specifying the trace files and nam files to be created.
- 3) Define a finish procedure.
- 4) Determine and create the nodes to be used for topology. Here we select 6 nodes: 0,1,2,3,4,5.
- 5) Create links for connecting these nodes.
- 6) Set up the LAN by specifying nodes and assign values for bandwidth, delay, queue type and channel to it.
- 7) Set up the TCP and UDP connection(s) and the FTP/CBR (or any other application) that will run over it.
- 8) Schedule the different events like simulation start and stop, data transmission start and stop.
- 9) Call the finish procedure and mention the time of end of simulation.
- 10) Execute the script in terminal by command: **ns script_name.tcl**

Program Code:

```
#lan.tcl
#Lan simulation
set ns [new Simulator]
#define color for data flows
$ns color 1 Blue
$ns color 2 Red
#open tracefiles
set tracefile1 [open out.tr w]
set winfile [open winfile w]
$ns trace-all $tracefile1
#open nam file
set namfile [open out.nam w]
$ns namtrace-all $namfile
#define the finish procedure
proc finish {} {
    global ns tracefile1 namfile
    $ns flush-trace
    close $tracefile1
    close $namfile
    exec nam out.nam &
    exit 0
}
```

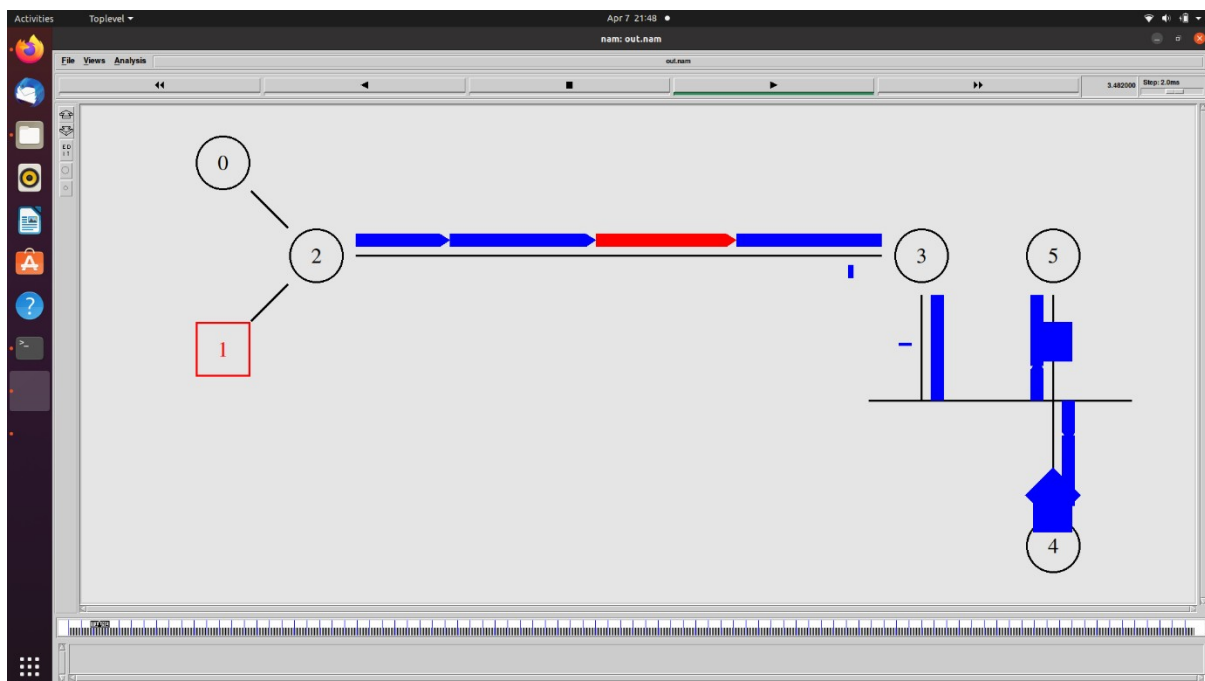
```

#create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color Red
$n1 shape box
#create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Cd Channel]
#Give node position
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left
#set queue size of link(n2-n3) to 20
$ns queue-limit $n2 $n3 20
#setup TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set packet_size_ 552
#set ftp over tcp connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
#setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 2
#setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR

```

```
$cbr set packet_size_ 1000
$cbr set rate_ 0.01Mb
$cbr set random_ false
#scheduling the events
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
$ns at 125.5 "$cbr stop"
proc plotWindow {tcpSource file} {
    global ns
    set time 0.1
    set now [$ns now]
    set cwnd [$tcpSource set cwnd_]
    puts $file "$now $cwnd"
    $ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
$ns at 0.1 "plotWindow $tcp $winfile"
$ns at 125.0 "finish"
$ns run
```

Output:



EXPERIMENT-11

Aim: Implementation of various MAC protocols.

Algorithm:

- 1) Create a Simulator object.
- 2) Define different colours for different data flows.
- 3) Open a nam trace file and define procedure, after which close the trace file and execute nam on trace file.
- 4) Create six nodes and form a network, 0,1,2,3,4,5.
- 5) Create duplex links between the nodes and add orientation to the nodes for topology.
- 6) Setup TCP connection between n(0) and n(4).
- 7) Apply FTP traffic over TCP.
- 8) Setup UDP Connections between n(1) and n(5).
- 9) Apply CBR traffic over UDP.
- 10) Apply CSMA/CA and CSMA/CD mechanisms and study their performance
- 11) Schedule the events and run the program.

Program Code:

```
#csma.tcl
set ns [new Simulator]
#Define different colors for data flows (for nam)
$ns color 1 Blue
$ns color 2 red
#Open the Trace files
set file1 [open out.tr w]
set winfile [open Winfile w]
$ns trace-all $file1
#Open the NAM trace file
set file2 [open out.nam w]
$ns namtrace-all $file2
#Define a 'finish' procedure
proc finish {} {
    global ns file1 file2
    $ns flush-trace
    close $file1
    close $file2
    exec nam out.nam &
    exit 0
}
#create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
```



```

set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color Red
$n1 shape box
#create link between nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/Droptail MAC/Csma/Ca Channel]
#setup a TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
#setup FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 2
#setup a cbr over udp connexion
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 0.01mb
$cbr set random_ false
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 123.0 "$ftp stop"
$ns at 124.5 "$cbr stop"
#next procedure gets two arguments: the name of tcp source node will br called here tcp
#and the name of output file

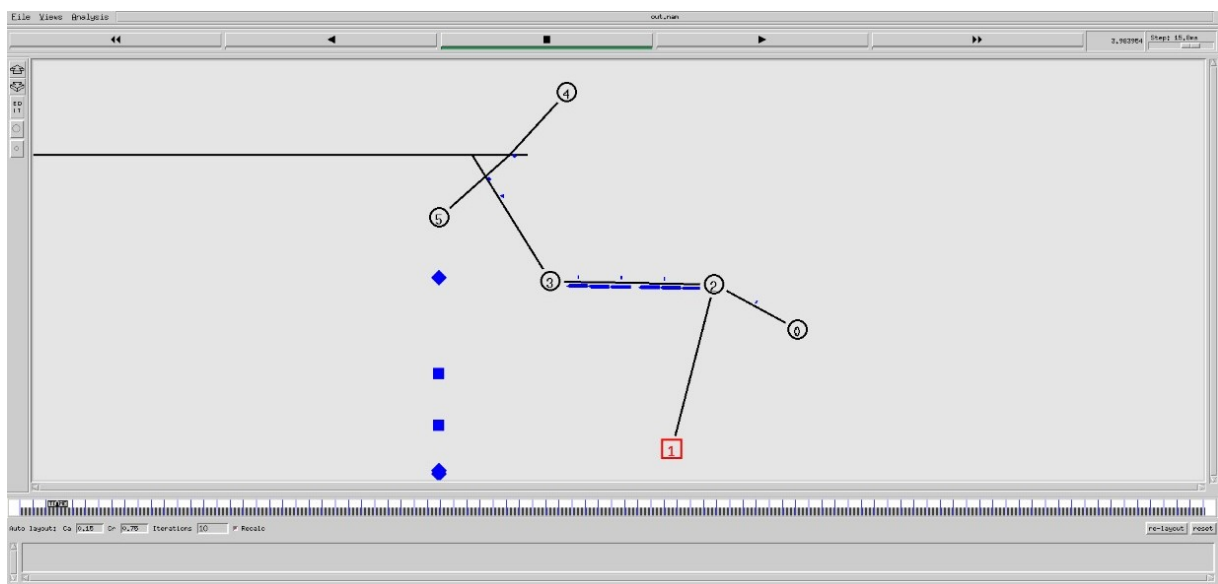
```

```

proc plotWindow {tcpSource file} {
  global ns
  set time 0.1
  set now [$ns now]
  set cwnd [$tcpSource set cwnd_]
  set wnd [$tcpSource set window_]
  puts $file "$now $cwnd"
  $ns at [expr $now+$time] "plotWindow $tcpSource $file"
  $ns at 0.1 "plotWindow $tcp $winfile"
  $ns at 5 "$ns trace-annotate \"packet drop\""
  #PPP
  $ns at 125.0 "finish"
  $ns run
}

```

Output:



EXPERIMENT-12

Aim: Measuring Network Performance: Network Performance Evaluation, Performance Evaluation Metrics.

Network Performance:

- Network Performance can be defined as the overall quality of services provided by the network.
- It is defined by various parameters and metrics that are analyzed together to assess the given network.
- There set of processes and tools that can be used for assessing the network performance quantitatively and qualitatively and provide actionable data to remediate any network performance issues that may arise.

Why network performance measurement is important?

- Network demands are increasing day by day, so there is need of good network performance.
- User satisfaction.
- Detect and diagnose performance issues.
- To ensure network quality.

Performance Evaluation Metrics:

- The choice of metric depends on the purpose of the network has been setup for, for example TCP throughput is based for application layer ad not network layer.
- Once these metrics are chosen, one goes for the qualitative evaluation by subjecting the network under the diverse conditions for example increasing the bandwidth of the links step by step.

The table below shows the evaluation metrics and the categories that they are appropriate for:

Category	Metric	Unit
Productivity	Throughput	Bytes per second
Responsiveness	Delay, jitter	Seconds
Utilization	Channel Utilization	Percentage of time busy
Loss	Packet drops, Retransmission count	Number
Buffer Space	Queue size, overflow and underflow rate	Bytes

However it might always be not feasible to obtain the best performance from network due to factors like high cost, complexity, compatibility. In such cases some factors are balanced for increasing the performance

Some performance measurement metrics are:

- 1) *Latency*: It might take a long time for a packet to be delivered across the networks. In reliable protocols a receiver acknowledges delivery of each chunk of data and thus it is possible to measure time as round-trip time.
- 2) *Packet Loss*: In some cases intermediate devices might lose some network due to some errors.
- 3) *Retransmission*: If packets are lost in a reliable network they are retransmitted. This results in two delays: first is the delay from re-sending of the data and second delay is resulted in the waiting until the data is received finally in the right order.
- 4) *Throughput*: The amount of traffic a network can carry is measured in throughput (Kbps).

Latency is analogous to the speed limit and throughput is analogous to number of lanes in the highway.

Latency = Propagation Time + Transmission Time + Queuing Time + Processing Delay

Propagation Time = Distance/Propagation Speed

Parameters affecting the Performance of Network:

Different parameters that affect the network performance are:

- 1) *Bandwidth*: It is defined as the maximum data transfer rate which a link allows and is expressed in nits per second (bps).
- 2) *Propagation Delay*: It is the amount of time required for a packet to travel from one node to another. If it is high, the throughput would reduce and vice versa.
- 3) *Queue type and queue size*: The queue of nodes is implemented as a part of the link whose input is that node to handle the overflow at the queue. But if the buffer capacity of output queue is exceeded then the last packet arrived would be dropped. This buffer capacity is set by queue size.

EXPERIMENT-13

Aim: Performance Evaluation of routing Protocol.

Routing Protocol:

This protocol is a set of rules used by the routers to communicate between the source and destination. Each protocol has its own algorithm to choose path.

There are two kinds of routing:

- 1) *Unicast Routing*: It is a type of information transfer and used when there is a participation of single sender and a single recipient.
- 2) *Multicast Routing*: It used in case of multiple senders and recipients.

Performance Analysis of routing Protocol:

The analysis of routing Protocol can be assessed by factors such as:

- 1) *Packet delivery Ratio*: It is measured as the ratio of number of packets delivered in total to the number of packets sent from the source node to destination node.
- 2) *Energy Consumptions*: Energy consumption is measured as the total energy used to perform the complete process.
- 3) *Throughput*: The number of packets transferred per unit time.
- 4) *Average Delay*: It is the delay per packet transfer from source node to the destination node.

All of the above factors and some others affect the performance of routing protocol and can be used to assess its performance.

Program Code:

The following program shows how an XGraph can be used to plot the bandwidth of two nodes connected through the duplex wired link (An XGraph program draws a graph on an x-display given data read from either data file or standard input):

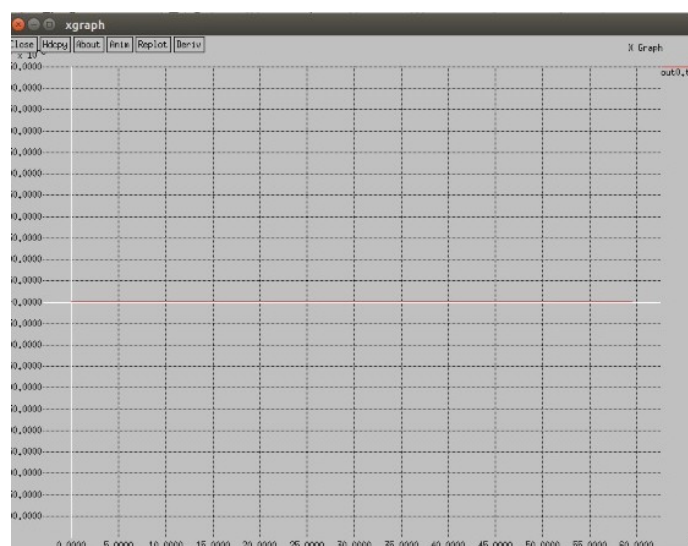
```
#Create a simulator object
set ns [new Simulator]
#Open the output trace file
set f0 [open out0.tr w]
#Create 2 nodes
set n0 [$ns node]
set n1 [$ns node]
#Connect the nodes using duplex link
$ns duplex-link $n0 $n1 1Mb 100ms DropTail
#Define a 'finish' procedure
proc finish {} {
    global f0
    #Close the output files
    close $f0
    #Call xgraph to display the results
```

```

exec xgraph out0.tr -geometry 800x400 &
exit 0
}
#Define a procedure which periodically records the bandwidth received by the
proc record {} {
global sink0 f0
#Get an instance of the simulator
set ns [Simulator instance]
#Set the time after which the procedure should be called again
set time 0.5
#How many bytes have been received by the traffic sinks?
set bw0 [$sink0 set bytes_]
#Get the current time
set now [$ns now]
#Calculate the bandwidth (in MBit/s) and write it to the files
puts $f0 "$now [expr $bw0/$time*8/1000000]"
#Reset the bytes_ values on the traffic sinks
$sink0 set bytes_ 0
#Re-schedule the procedure
$ns at [expr $now+$time] "record"
}
#Create three traffic sinks and attach them to the node n4
set sink0 [new Agent/LossMonitor]
#Start logging the received bandwidth
$ns at 0.0 "record"
$ns at 60.0 "finish"
#Run the simulation
$ns run

```

Output:



EXPERIMENT-14

Aim: Parameter Affecting the Performance of Network, Performance Evaluation Technique, Network Performance Evaluation using NS-2/3

Network Performance:

Performance of network is the measurement of the quality of services provided by the network as viewed by the customer. Performance evaluation is the application of the scientific method to the study of computer systems. It is very important to measure the network performance, so as to improve and maintain quality.

Parameters affecting performance of network:

There are various factors that affect the performance of a network, some of these are:

- 1) *Network Latency*: It refers to the time needed to send a packet from source to the destination and can be dependent on:
 - Physical distances between source and destination.
 - Number of network devices involved.
 - Performance of each device.
 - Protocol used.
- 2) *Network Congestion*: It refers to the saturation of path used the packets to flow from the source to the destination. When the maximum capacity is reached, a congestion occurs and it may cause time delays.
- 3) *Infrastructure Parameters*: There might be some infrastructure rules that might affect the network performance, some of which are:
 - Prioritization
 - Filtering
 - Encryption
 - Load Balancing
- 4) *Bandwidth*: It is defined as the maximum data transfer rate which a link allows and is expressed in nits per second (bps), and affects the maximum amount of packets that can be transferred from source to the destinantion.
- 5) *Propagation Delay*: It is the amount of time required for a packet to travel from one node to another. If it is high, the throughput would reduce and vice versa. Propagation delay can be caused of many factors that might affect the transmission.
- 6) *Queue type and queue size*: The queue of nodes is implemented as a part of the link whose input is that node to handle the overflow at the queue. But if the buffer capacity of output queue is exceeded then the last packet arrived would be dropped. This buffer capacity is set by queue size.

Performance Evaluation Techniques:

The performance of a network is evaluated by a team through various techniques and strategy to keep up with the quality of the network. Performance evaluation techniques have been developed to accurately measure the effectiveness with which computer system resources are managed while striving to provide service that is fair to all customer classes. Some of these techniques are:

- 1) *Workload Characterization*: It involves the design and choice of traffic types that provides input for evaluation of computer network performance. Performance measures of computer networks are all dependent to some extent on the input workload, the network topology and the choices in controlled parameters or network default settings. In this evaluation, performance is measured under a given traffic workload and network configuration.
- 2) *Simulation Models*: They reproduce the behaviour of the network in a time domain, and defines it in terms of states and transitions. It is a numeric solution taken out by simulating to capture the behaviour of network in terms of logical condition by utilizing system of equations and data structures. Its three kinds are:
 - Trace-driven
 - Program-driven
 - Distribution-driven
- 3) *Analytic Model*: They involve system of equations (as simulation models), however they start with a network of queues model and develop a system of equations that may be used to get a closed form solution. They are based on the theory of stochastic processes associated with the independent random variables.

Network Performance Evaluation using NS-2/3

NS stands for Network Simulator. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. They come under Simulation Models and involve analyzing the performance by considering memory usage and runtime metrics. We also compare the average throughput, the average received rate and number of received packets for different area sizes and different number of nodes.

EXPERIMENT-15

Aim: Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

Program Code:

```
set ns [new Simulator]
set tf [open lab7.tr w]
$ns trace-all $tf
set nf [open lab7.nam w]
$ns namtrace-all $nf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns make-lan "$n0 $n1 $n2 $n3" 10mb 10ms LL Queue/DropTail Mac/802_3
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp0 $sink3
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp2 $sink1
#####To trace the congestion window#####
set file1 [open file1.tr w]
$tcp0 attach $file1
$tcp0 trace cwnd_
$tcp0 set maxcwnd_ 10
set file2 [open file2.tr w]
$tcp2 attach $file2
$tcp2 trace cwnd_
proc finish { } {
    global nf tf ns
    $ns flush-trace
    exec nam lab7.nam &
    close $nf
    close $tf
```

```

    exit 0
}
$ns at 0.1 "$ftp0 start"
$ns at 1.5 "$ftp0 stop"
$ns at 2 "$ftp0 start"
$ns at 3 "$ftp0 stop"
$ns at 0.2 "$ftp2 start"
$ns at 2 "$ftp2 stop"
$ns at 2.5 "$ftp2 start"
$ns at 4 "$ftp2 stop"
$ns at 5.0 "finish"
$ns run

```

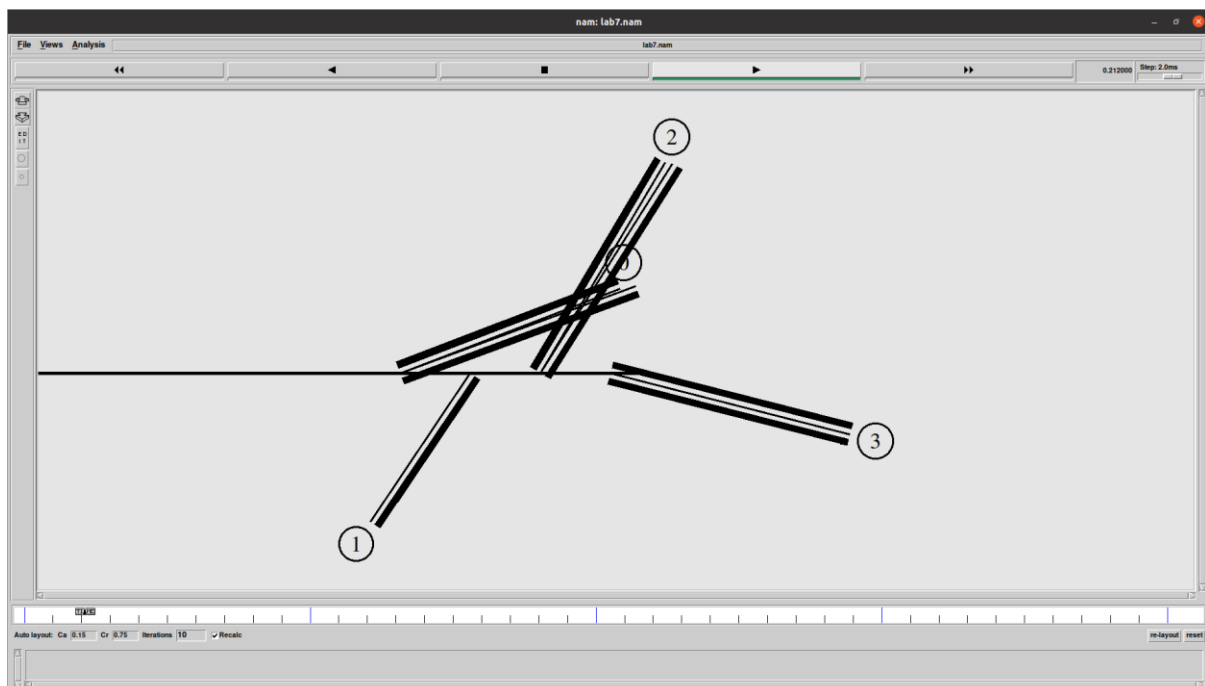
awk script:

```

BEGIN {
}
{
if($6=="cwnd_") /* don't leave space after writing cwnd_ */
printf("%f\t%f\t\n",$1,$7); /* you must put \n in printf */
}
END {
}

```

Output:



EXPERIMENT-16

Aim: Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.

Program Code:

```
set ns [new Simulator]
set tf [open exp4.tr w]
$ns trace-all $tf
set topo [new Topography]
$topo load_flatgrid 1000 1000
set nf [open exp4.nam w]
$ns namtrace-all-wireless $nf 1000 1000
$ns node-config -adhocRouting DSDV \
    -llType LL \
    -macType Mac/802_11 \
    -ifqType Queue/DropTail \
    -ifqLen 50 \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -propType Propagation/TwoRayGround \
    -antType Antenna/OmniAntenna \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON
create-god 3
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$n0 label "tcp0"
$n1 label "sink1/tcp1"
$n2 label "sink2"
#The below code is used to give the initial node positions.
$n0 set X_ 50
$n0 set Y_ 50
$n0 set Z_ 0
$n1 set X_ 100
$n1 set Y_ 100
$n1 set Z_ 0
$n2 set X_ 600
$n2 set Y_ 600
$n2 set Z_ 0
$ns at 0.1 "$n0 setdest 50 50 15"
```

```

$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp1 $sink2
$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"
#The below code is used to provide the node movements.
$ns at 100 "$n1 setdest 550 550 15"
$ns at 190 "$n1 setdest 70 70 15"
proc finish {} {
    global ns nf tf
    $ns flush-trace
    exec nam exp4.nam &
    close $tf
    exit 0
}
$ns at 250 "finish"
$ns run

```

awk script:

```

BEGIN{
#include<stdio.h>
count1=0
count2=0
pack1=0
pack2=0
time1=0
time2=0

}
{

```

```

if($1=="r"&&$3=="_1_"&&$4=="AGT")
{
count1++
pack1=pack1+$8
time1=$2
}
if($1=="r"&&$3=="_2_"&&$4=="AGT")
{

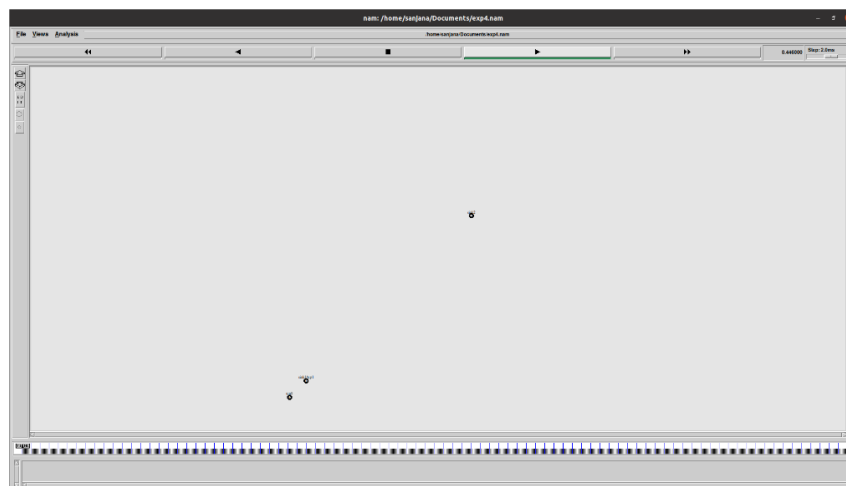
}
}
END{

count2++
pack2=pack2+$8
time2=$2

printf("The Throughput from n0 to n1: %fMbps\n",
((count1*pack1*8)/(time1*1000000)))
printf("The Throughput from n1 to n2: %f Mbps\n",
((count2* pack2 * 8) /(time2*1000000)))
}

```

Output:



EXPERIMENT-17

Aim: Write a program for error detecting code using CR C-C CITT (16- bits)

Program Code:

```
package lab;
import java.io.*;

public class Experiment17 {
    public static void main(String args[]) throws IOException
    {
        System.out.println("Roll number: 19115081 \n");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        int[ ] data;
        int[ ]div;
        int[ ]divisor;
        int[ ]rem;
        int[ ] crc;
        int data_bits, divisor_bits, tot_length;
        System.out.println("Enter number of data bits: ");
        data_bits=Integer.parseInt(br.readLine());
        data=new int[data_bits];
        System.out.println("Enter data bits: ");
        for(int i=0; i<data_bits; i++)
            data[i]=Integer.parseInt(br.readLine());
        System.out.println("Enter number of bits in divisor: ");
        divisor_bits=Integer.parseInt(br.readLine());
        divisor=new int[divisor_bits];
        System.out.println("Enter Divisor bits: ");
        for(int i=0; i<divisor_bits; i++)
            divisor[i]=Integer.parseInt(br.readLine());
        tot_length=data_bits+divisor_bits;
        div=new int[tot_length];
        rem=new int[tot_length];
        crc=new int[tot_length];
        /*----- CRC GENERATION -----*/
        for(int i=0;i<data.length;i++)
            div[i]=data[i];
        System.out.print("Dividend (after appending 0's) are: ");
        for(int i=0; i< div.length; i++)
            System.out.print(div[i]);
        System.out.println();
        for(int j=0; j<div.length; j++){
            rem[j] = div[j];
        }
        rem=divide(div, divisor, rem);
        for(int i=0;i<div.length;i++)
            //append dividend and remainder
```

```

    {
        crc[i]=(div[i]^rem[i]);
    }
    System.out.println();
    System.out.println("CRC code : ");
    for(int i=0;i<crc.length;i++)
        System.out.print(crc[i]);
    /*-----ERROR DETECTION -----*/
    System.out.println();
    System.out.println("\n Enter CRC code of "+ tot_length +" bits : ");
    for(int i=0; i<crc.length; i++)
        crc[i]=Integer.parseInt(br.readLine());
    for(int j=0; j<crc.length; j++){
        rem[j] = crc[j];
    }
    rem=divide(crc, divisor, rem);
    for(int i=0; i< rem.length; i++)
    {
        if(rem[i]!=0)
        {
            System.out.println("There is an error!");
            break;
        }
        if(i==rem.length-1)
            System.out.println("The code is error-free!");
    }
    System.out.println("THANK YOU.....");
}

static int[] divide(int div[],int divisor[], int rem[])
{
    int cur=0;
    while(true)
    {
        for(int i=0;i<divisor.length;i++)
            rem[cur+i]=(rem[cur+i]^divisor[i]);
        while(rem[cur]==0 && cur != rem.length-1)
            cur++;
        if((rem.length-cur)<divisor.length)
            break;
    }
    return rem;
}
}

```

Output:

I)

```
<terminated> Experiment17 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Apr 18, 2021, 6:36:43 PM – 6:36:58 PM)
Roll number: 19115081

Enter number of data bits:
4
Enter data bits:
1
1
1
0
Enter number of bits in divisor:
3
Enter Divisor bits:
1
1
0
Dividend (after appending 0's) are: 1110000

CRC code :
1110010

Enter CRC code of 7 bits :
1
1
1
0
0
1
0
The code is error-free!
THANK YOU.....
```

II)

```
<terminated> Experiment17 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Apr 18, 2021, 6:37:07 PM – 6:37:22 PM)
Roll number: 19115081

Enter number of data bits:
4
Enter data bits:
1
1
1
0
Enter number of bits in divisor:
3
Enter Divisor bits:
1
1
0
Dividend (after appending 0's) are: 1110000

CRC code :
1110010

Enter CRC code of 7 bits :
1
1
1
0
0
1
1
There is an error!
THANK YOU.....
```


EXPERIMENT-18

Aim: Write a program to find the shortest path between vertices using bellman-ford algorithm.

Program Code:

```
package lab;
import java.util.Scanner;

public class Experiment18 {
    private int D[];
    private int num_ver;
    public static final int MAX_VALUE = 999;
    public Experiment18(int num_ver)
    {
        this.num_ver = num_ver;
        D = new int[num_ver + 1];
    }
    public void BellmanFordEvaluation(int source, int A[][])
    {
        for (int node = 1; node <= num_ver; node++)
        {
            D[node] = MAX_VALUE;
        }
        D[source] = 0;
        for (int node = 1; node <= num_ver - 1; node++)
        {
            for (int sn = 1; sn <= num_ver; sn++)
            {
                for (int dn = 1; dn <= num_ver; dn++)
                {
                    if (A[sn][dn] != MAX_VALUE)
                    {
                        if (D[dn] > D[sn] + A[sn][dn])
                            D[dn] = D[sn] + A[sn][dn];
                    }
                }
            }
        }
        for (int sn = 1; sn <= num_ver; sn++)
        {
            for (int dn = 1; dn <= num_ver; dn++)
            {
                if (A[sn][dn] != MAX_VALUE)
                {
                    if (D[dn] > D[sn] + A[sn][dn])
                        System.out.println("The Graph contains a negative egde cycle!");
                }
            }
        }
    }
}
```

```

        for (int vertex = 1; vertex <= num_ver; vertex++)
        {
            System.out.println("Distance of source " + source + " to " + vertex + " vertex is " +
D[vertex]);
        }
    }
    public static void main(String[ ] args)
    {
        System.out.println("Roll number: 19115081 \n");
        int num_ver= 0;int source;
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of vertices in graph: ");
        num_ver = scanner.nextInt();
        int A[][] = new int[num_ver + 1][num_ver + 1];
        System.out.println("Enter the adjacency matrix of the graph: ");
        for (int sn = 1; sn <= num_ver; sn++)
        {
            for (int dn = 1; dn <= num_ver; dn++)
            {
                A[sn][dn] = scanner.nextInt();
                if (sn == dn)
                {
                    A[sn][dn] = 0;
                    continue;
                }
                if (A[sn][dn] == 0)
                {
                    A[sn][dn] = MAX_VALUE;
                }
            }
        }
        System.out.println("Source vertex: ");
        source = scanner.nextInt();
        Experiment18 b = new Experiment18(num_ver);
        b.BellmanFordEvaluation(source, A);
        scanner.close();
    }
}

```

Output:

```
<terminated> Experiment18 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Apr 18, 2021, 6:28:02 PM – 6:29:12 PM)
Roll number: 19115081

Enter the number of vertices in graph:
5
Enter the adjacency matrix of the graph:
0 2 4 999 999
4 0 3 3 5
2 3 0 999 2
999 3 999 0 2
999 5 2 2 0
Source vertex:
2
Distance of source 2 to 1 vertex is 4
Distance of source 2 to 2 vertex is 0
Distance of source 2 to 3 vertex is 3
Distance of source 2 to 4 vertex is 3
Distance of source 2 to 5 vertex is 5
```

EXPERIMENT-19

Aim: Write a program for simple RSA algorithm to encrypt and decrypt the data.

Program Code:

```
package lab;
import java.util.*;
import java.util.Scanner;

public class Experiment19 {
    public static void main(String arg[]){
        System.out.println("Roll number: 19115081 \n");
        Scanner in=new Scanner(System.in);
        long p,q,d,z,e,n,c;
        int choice;
        System.out.println("Enter two distinct prime numbers: ");
        p=in.nextLong();
        q=in.nextLong();
        n=p*q;
        z=(p-1)*(q-1);
        System.out.println("Enter a value of 'd' which is less than and relatively prime
to "+z + ": ");
        d=in.nextLong();
        for(e=1;e<z;++e)
        {
            if(((e*d)%z)==1)
                break;
        }

        System.out.println("p="+p+"\nq="+q+"\nn="+n+"\nz="+z+"\nd="+d+"\ne="+e+
"\n");

        do{
            System.out.println("1.Encryption \n2.Decryption\n3.Exit");
            System.out.println("Choose an option (1-3): ");
            choice = in.nextInt();
            switch(choice){
                case 1: System.out.println("Enter a plain text: ");
                    String s = in.next();
                    System.out.println("Plain Text" + "\t" + "Cipher Text");
                    for(int i = 0; i < s.length(); i++){
                        long pl = (int) s.charAt(i);
                        c = modexp(pl,e,n);
                        System.out.println(s.charAt(i)+"\t" + c);
                    }
                    break;
                case 2: System.out.println("Enter a cipher text (0 to stop input): ");
                    long [] ci = new long[50];
                    int j =0;
                    do{
```

```

        ci[j] = in.nextLong();
    } while (ci[j++] != 0);
    System.out.println("Cipher Text" + "\t" + "Plain Text");
    for (int i = 0; i < j-1; i++) {
        long pl = modexp(ci[i], d, n);
        System.out.println(ci[i] + "\t\t" + (char) pl);
    }
    break;
case 3: System.out.println("Program Terminated!!");
System.exit(0);
}
} while (choice != 3);
}

static long modexp(long a, long x, long n)
{
    long r = 1;
    while (x > 0)
    {
        if (x % 2 == 1) {
            r = (r * a) % n;
        }
        a = (a * a) % n;
        x /= 2;
    }
    return (r);
}
}

```

Output:

```

<terminated> Experiment19 [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Apr 18, 2021, 6:15:04 PM - 6:15:22 PM)
Roll number: 19115081

Enter two distinct prime numbers:
5 7
Enter a value of 'd' which is less than and relatively prime to 24:
11
p=5
q=7
n=35
z=24
d=11
e=11

1.Encryption
2.Decryption
3.Exit
Choose an option (1-3):
1
Enter a plain text:
Rain
Plain Text      Cipher Text
R                3
a                13
i                0
n                10
1.Encryption
2.Decryption
3.Exit
Choose an option (1-3):
3
Program Terminated!!

```

EXPERIMENT-20

Aim: Write a program for congestion control using leaky bucket algorithm.

Program Code:

```
package lab;
import java.io.*;
import java.util.*;
class Queue
{
int q[],f=0,r=0,size;
void insert(int n)
{
Scanner in = new Scanner(System.in);
q=new int[10];
for(int i=0;i<n;i++)
{
System.out.print("\nEnter " + i + " packet: ");
int ele=in.nextInt();
if(r+1>10)
{
System.out.println("\nQueue is full! \nLost Packet: "+ele);
break;
}
else
{
r++;
q[i]=ele;
}
}
}
void delete()
{
Scanner in = new Scanner(System.in);
Thread t=new Thread();
if(r==0)
System.out.print("\nQueue is empty ");
else
{
for(int i=f;i<r;i++)
{
try
{
t.sleep(1000);
}
catch(Exception e){}
System.out.print("\nLeaked Packet: "+q[i]); f++;
}
}
```

```
}  
System.out.println();  
}  
}
```

```
public class Experiment20 extends Thread {  
  
    public static void main(String ar[]) throws Exception {  
        Queue q=new Queue();  
        Scanner src =new Scanner(System.in);  
        System.out.println("Roll number: 19115081\n");  
        System.out.println("\nEnter the total number of packets to be sent: ");  
        int size=src.nextInt();  
        q.insert(size);  
        q.delete();  
    }  
  
}
```

Output:

```
Roll number: 19115081  
  
Enter the total number of packets to be sent:  
5  
Enter 0 packet: 1  
Enter 1 packet: 4  
Enter 2 packet: 7  
Enter 3 packet: 10  
Enter 4 packet: 12  
|  
Leaked Packet: 1  
Leaked Packet: 4  
Leaked Packet: 7  
Leaked Packet: 10  
Leaked Packet: 12
```

```
# lan.tcl
# Lan simulation
set ns [new Simulator]
# define color for data flows
$ns color 1 Blue
$ns color 2 Red.
# open tracefiles
set tracefile [open out.tr.w]
set winfile [open winfile.w]
$ns trace-all $tracefile
# open nam file
set namfile [open out.nam w].
$ns namtrace-all $namfile
# define the finish procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam.out.nam &
    exit()
}

# create six nodes.
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color Red
$n1 shape box

# create links between the nodes.
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
```


10
\$ns simplex-link \$n2 \$n3 0.3 Mb 100 ms DropTail.

\$ns simplex-link \$n3 \$n2 0.3 Mb 100 ms DropTail.

set lan [\$ns newlan "\$n3 \$n4 \$n5 0.5 Mb 40 ms LL Queue/DropTail mac
/csma/cd channel],

Give node position.

\$ns duplex-link-op \$n0 \$n2 orient right-down.

\$ns duplex-link-op \$n1 \$n2 orient right-up.

\$ns simplex-link-op \$n2 \$n3 orient right

\$ns simplex-link-op \$n3 \$n2 orient left.

Set queue size of link(n2-n3) to 20.

\$ns queue-limit \$n2 \$n3 20.

Setup TCP connection.

set tcp [new Agent/TCP/Newreno].

\$ns attach-agent \$n4 \$sink.

\$ns connect \$tcp \$sink.

\$tcp set jid-1

\$tcp set packet-size-852

Set ftp over tcp connection.

set ftp [new Application/FTP].

\$ftp attach-agent \$tcp.

Setup a UDP connection.

set udp [new Agent/UDP].

\$ns attach-agent \$n1 \$udp.

set null [new Agent/Null]

\$ns attach-agent \$n5 \$null

\$ns connect \$udp \$null.

\$udp set jid-2

Setup a CBR over UDP connection.

set cbr [new Application/Traffic/CBR].

\$cbr attach-agent \$udp.

\$cbr set type - CBR.

\$cbr set packet-size 1000 .

\$cbr set rate 0.01 Mb .

\$cbr set random false

scheduling the elements .

\$ns at 0.1 "\$cbr start"

\$ns at 1.0 "\$dtp start" .

\$ns at 124.0 "\$dtp stop" .

\$ns at 125.5 "\$cbr stop" .

proc plotWindow {tcpSource file} {

global ns

set time 0.1

set now [\$ns now]

set cwnd [\$tcpSource get cwnd-].

puts \$file "\$now \$cwnd"

\$ns at [expr \$now + \$time] "plotWindow \$tcpSource \$file"

}

\$ns at 0.1 "plotWindow \$tcp \$winfile".

\$ns at 125.0 "finish" .

\$ns run .

```
# csma.tcl
```

```
set ns [new Simulator].
```

```
# Define different colors for data flows (for nam)
```

```
$ns color 1 Blue
```

```
$ns color 2 red
```

```
# Open the Trace files
```

```
set file1 [open out.tr w].
```

```
set winfile [open winfile.w].
```

```
$ns trace-all $file1
```

```
# Open the NAM trace file.
```

```
set file2 [open out.nam w].
```

```
$ns namtrace-all $file2.
```

```
# Define a "finish" procedure
```

```
proc finish {} {
```

```
    global ns file1 file2
```

```
    $ns flush-trace.
```

```
    close $file1
```

```
    close $file2
```

```
    exec nam out.nam &
```

```
    exit 0
```

```
}
```

```
# create six nodes
```

```
set no. [$ns node].
```

```
set n1 [$ns node].
```

```
set n2 [$ns node]
```

```
set n3 [$ns node].
```

```
set n4 [$ns node].
```

```
set n5 [$ns node]
```

```
$n1 color Red-
```

```
$n1 shape box
```

```
# create link between nodes
```

```
$ns duplex-link $no $n2 2Mb 10ms DropTail.
```

```
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
```

```
$ns simplex-link $n2 $n3 0.3 Mb 100ms DropTail
```

```
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail.
```

```
set lan [$ns newLan "$n3 $n4 $n5" 0.5 Mb 40ms LL Queue/DropTail MAC/
      CsmA/Ca channel]
```

```
# Setup a TCP connection.
```

```
set tcp [new Agent/TCP/New-reno].
```

```
$ns attach-agent $n0 $tcp.
```

```
set sink [new Agent/TCPsink/DelAck].
```

```
$ns attach-agent $n4 $sink.
```

```
$ns connect $tcp $sink
```

```
$tcp set jid_ 1
```

```
$tcp set window_ 8000
```

```
$tcp set packetSize_ 552.
```

```
# Setup FTP over TCP connection.
```

```
set ftp [new Application/FTP].
```

```
$ftp attach-agent $tcp.
```

```
$ftp set-type_ FTP.
```

```
# Setup a UDP connection.
```

```
set udp [new Agent/UDP].
```

```
$ns attach-agent $n1 $udp.
```

```
set null [new Agent/Null].
```

```
$ns attach-agent $n5 $null.
```

```
$ns connect $udp $null
```

```
$udp set jid_ 2
```

```
# Setup a cbr over udp connection.
```

```
set cbr [new Application/Traffic/CBR].
```

```
$cbr attach-agent $udp
```

```
$cbr set type_ CBR.
```

```
$cbr set packet-size_ 1000.
```

```
$cbr set rate_ 0.01 mb.
```

```
$cbr set random_ false
```

```
$ns at 0.1 "$cbr start"
```

```
$ns at 1.0 "$ftp start".
```


\$ns at 123.0 "\$ftp stop".

\$ns at 124.5 "\$cbr stop".

next procedure gets two arguments: the name of tcp source node
will be called here tcp and the name of output file.

```
proc plotWindows {tcpSource file} {
```

```
  global ns
```

```
  set time 0.1
```

```
  set now [$ns now].
```

```
  set cwnd [$tcpSource set cwnd].
```

```
  set wnd [$tcpSource set window- ] .
```

```
  puts $file "$now $cwnd".
```

```
  $ns at [expr $now + $time] "plotWindow $tcpSource $file"
```

```
}
```

```
$ns at 0.1 "plotWindow $tcp $winfile".
```

```
$ns at 5 "$ns trace annotate \"packet drop\""
```

```
# PPP.
```

```
$ns at 125.0 "finish".
```

```
$ns run.
```

```
# Create a simulator object
```

```
set ns [new Simulator].
```

```
# Open the output trace file.
```

```
set fo [open out0.tr w].
```

```
# Create 2 nodes.
```

```
set n0 [$ns node]
```

```
set n1 [$ns node].
```

```
# connect the nodes using duplex link.
```

```
$ns duplex-link $n0 $n1 1mb 100ms DropTail.
```

```
# Define a 'finish' procedure.
```

```
proc finish {} {
```

```
    global fo.
```

```
    # Close the output files.
```

```
    close $fo.
```

```
    # Call xgraph to display the results.
```

```
    exec xgraph out0.tr -geometry 800x400 &.
```

```
    exit 0.
```

```
}
```

```
# Define a procedure which periodically records the bandwidth  
# received by the
```

```
proc record {} {
```

```
    global sink0 fo.
```

```
    # get an instance of the simulator.
```

```
    set ns [Simulator instance].
```

```
    # Set the time after which the procedure should be called  
    # again.
```

```
    set time 0.5.
```

```
    # How many time bytes have been received by the traffic sinks.
```

```
    set bwo [$sink0 set bytes]
```

```
    # Get the current time
```

```
    set now [$ns now].
```

```
    # Calculate the bandwidth (in Mbit/s) and write it to the files
```

```
    puts $fo " $now [expr $bwo / $time * 8 / 1000000]"
```

Reset the bytes - values on the traffic sinks.

\$ sink 0 set bytes - 0.

Reschedule the procedure.

\$ns at [expr \$now + \$time] "record".

}

Create three traffic sinks and attach them to the node n4.

Set sink 0. [new Agent/Loss Monitor],

Start logging the received bandwidth.

\$ns at 0.0 "record".

\$ns at 60.0 "finish".

Run the simulation.

\$ns run.

```

set ns [new Simulator] .
set tf [open lab7.tr w]
$ns trace-all $tf .
set nf [open lab7.nam w] .
$ns namtrace-all $nf.
set no [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node].
$ns make-lan "$no $n1 $n2 $n3" 10mb 10ms 2 Queue/DropTail Mac/
802.3.
set tcp0 [new Agent/TCP] .
$ns attach-agent $no $tcp0.
set ftp0 [new Application/FTP] .
$ftp0 attach-agent $tcp0.
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3.
$ns connect $tcp0 $sink3.
set tcp2 [new Agent/TCP] .
$ns attach-agent $n2 $tcp2.
set ftp2 [new Application/FTP] .
$ftp2 attach-agent $tcp2.
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1.
$ns connect $tcp2 $sink1.
## To trace the congestion window ##
set file1 [open file1.tr w]
$tcp0 attach $file1
$tcp0 trace cwnd.
$tcp0 set maxcwnd 10.
set file2 [open file2.tr w] .
$tcp2 attach $file2.

```


\$tcp2 trace cwnd_

proc finish {} {

global nf tf ns.

\$ns flush-trace.

exec nam. lab7.nam &.

close \$nf

close \$tf.

exit 0.

}

\$ns at 0.1 "\$tcp0 start".

\$ns at 1.5 "\$tcp0 stop".

\$ns at 2 "\$tcp0 start".

\$ns at 3 "\$tcp0 stop".

\$ns at 0.2 "\$tcp2 start".

\$ns at 2 "\$tcp2 stop".

\$ns at 2.5 "\$tcp2 start".

\$ns at 4 "\$tcp2 stop".

\$ns at 5.0 "finish".

\$ns run.

awk script:

BEGIN {

{

{

if (\$6 == "cwnd_") /* don't leave space after writing cwnd_ */

printf ("%d\t%d\t\n", \$1, \$7)

}

END {

}

Practical 16

```

set ns [new Simulator]
set tf [open exp4.tr w]
$ns trace-all $tf
set topo [new Topography]
$topo load-Hatgrid 1000 1000
set nf [open exp4.nam w]
$ns namtrace-all-wireless $nf 1000 1000
$ns node-config -adhocRouting OSDV \
    -llType LL \
    -macType Mac802_11 \
    -ifqType Queue/DropTail \
    -ifqlen 50 \
    -phyType Phy/WirelessChannel \
    -propType Propagation/TwoRayGround \
    -antType Antenna/OmniAntenna \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON

```

create-god 3.

set no [\$ns node]

set n1 [\$ns node]

set n2 [\$ns node]

\$no label "tcp0"

\$n1 label "sink1/tcp1"

\$n2 label "sink2"

The below code is used to give the initial node positions

\$no set x- 50

\$n1 set y- 50

\$no set z- 0

\$n1 set x- 100

\$n1 set y- 100

\$n1 set z- 0

\$n2 set x- 600

```

$ns set y_ 600.
$ns set z_ 0.
$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"
set tcp0 [new Agent /TCP].
$ns attach-agent $n0 $tcp0.
set ftp0 [new Application /FTP].
$ftp0 attach-agent $tcp0.
set sink1 [new Agent /TCPSink].
$ns attach-agent $n1 $sink1.
$ns connect $tcp0 $sink1.
set tcp1 [new Agent /TCP].
$ns attach-agent $n1 $tcp1.
set ftp1 [new Application /FTP].
$ftp1 attach-agent $tcp1.
set sink2 [new Agent /TCPSink].
$ns attach-agent $n2 $sink2.
$ns connect $tcp1 $sink2.
$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"
# The below code is used to provide the node movements.
$ns at 100 "$n0 setdest 550 550 15"
$ns at 190 "$n1 setdest 70 70 15"
proc finish {} {
    global ns nf tf.
    $ns flush-trace
    exec nam exp4.nam &
    close $tf.
    exit 0.
}
$ns at 250 "finish"
$ns run.

```


16
awk script :

BEGIN {

#include <stdio.h>

count1 = 0

count2 = 0

pack1 = 0

pack2 = 0

time1 = 0

time2 = 0

}

{

if (\$1 == "r" && \$3 == "-1-" && \$4 == "AGT")

{

count1++

pack1 = pack1 + \$8

time1 = \$2

}

if (\$1 == "r" && \$3 == "-2-" && \$4 == "AGT")

{

}

}

END {

count2++

pack2 = pack2 + \$8

time2 = \$2

printf ("The throughput from n0 to n1 : %f Mbps \n",

((count1 * pack1 * 8) / (time1 * 1000000))

printf ("The throughput from n1 to n2 : %f Mbps \n",

((count2 * pack2 * 8) / (time2 * 1000000)))

}

```
package lab;
```

```
import java.io.*;
```

```
public class Experiment 17 {
```

```
    public static void main (String args[]) throws IOException {
```

```
        System.out.println ("Roll no: 20115903");
```

```
        BufferedReader br = new BufferedReader (new InputStreamReader  
                                                    (System.in));
```

```
        int [] data;
```

```
        int [] div;
```

```
        int [] divisor, rem, crc;
```

```
        int data-bits, divisor-bits, tot-length;
```

```
        System.out.println ("Enter the number of data bits: ");
```

```
        data-bits = Integer.parseInt (br.readLine());
```

```
        data = new int [data-bits];
```

```
        System.out.println ("Enter data bits ");
```

```
        for (int i=0; i<data-bits; i++){
```

```
            data [i] = Integer.parseInt (br.readLine());
```

```
        }
```

```
        System.out.println ("Enter number of bits in divisor:");
```

```
        divisor-bits = Integer.parseInt (br.readLine());
```

```
        divisor = new int [divisor-bits];
```

```
        System.out.println ("Enter Divisor bits: ");
```

```
        for (int i=0; i<divisor-bits; i++){
```

```
            divisor [i] = Integer.parseInt (br.readLine());
```

```
        }
```

```
        tot-length = data-bits + divisor-bits;
```

```
        div = new int [tot-length];
```

```
        rem = new int [tot-length];
```

```
        crc = new int [tot-length];
```

```
        // crc generation.
```

```

for (int i=0; i<data.length; i++){
    div[i] = data[i];
}
System.out.println("Dividend after appending 0's are: ");
for (int i=0; i<div.length; i++){
    System.out.print(div[i]);
}
System.out.println();
for (int j=0; j<div.length; j++){
    rem[j] = div[j];
}
rem = divide(div, divisor, rem);
for (int i=0; i<div.length; i++){
    crc[i] = (div[i] ^ rem[i]);
}
System.out.println();
System.out.println("\n Enter crc code of "+ tot_length + " bits: ");
for (int i=0; i<crc.length; i++){
    crc[i] = Integer.parseInt(br.readLine());
}
for (int j=0; j<crc.length; j++){
    rem[j] = crc[j];
}
rem = divide(crc, divisor, rem);
for (int i=0; i<rem.length; i++){
    if (rem[i] != 0){
        System.out.println("There is an error");
        break;
    }
    if (i == rem.length - 1){
        System.out.println("The code is error free");
    }
}
System.out.println("THANK YOU....");
}

```

```
static int[] divide (int div[], int divisor[], int rem[])
```

```
int cur=0;
```

```
while (true){
```

```
    for (int i=0; i<divisor.length; i++){
```

```
        rem[cur+i] = (rem[cur+i] ^ divisor[i]);
```

```
    }
```

```
    while (rem[cur]==0 && cur!=rem.length-1){
```

```
        cur++;
```

```
    }
```

```
    if ((rem.length - cur) < divisor.length){
```

```
        break;
```

```
    }
```

```
}
```

```
return rem;
```

```
}
```


Practical 18

```

package lab;
import java.util.Scanner;

public class Experiment18 {
    private int D[];
    private int num-ver;
    public static final int MAX-VALUE = 999;
    public Experiment18(int num-ver) {
        this.num-ver = num-ver;
        D = new int [num-ver + 1];
    }
    public void BellmanFordEvaluation(int source, int A[][]) {
        for (int node = 1; node <= num-ver; node++) {
            D[node] = MAX-VALUE;
        }
        D[source] = 0;
        for (int node = 1; node <= num-ver - 1; node++) {
            for (int sn = 1; sn <= num-ver; sn++) {
                for (int dn = 1; dn <= num-ver; dn++) {
                    if (A[sn][dn] != MAX-VALUE) {
                        if (D[dn] > D[sn] + A[sn][dn]) {
                            D[dn] = D[sn] + A[sn][dn];
                        }
                    }
                }
            }
        }
        for (int sn = 1; sn <= num-ver; sn++) {
            for (int dn = 1; dn <= num-ver; dn++) {
                if (A[sn][dn] != MAX-VALUE) {
                    if (D[sn] > D[dn] + A[sn][dn]) {
                        System.out.println("The graph contains negative edge cycle");
                    }
                }
            }
        }
    }
}

```



```

    }
}
for (int vertex = 1; vertex <= num_ver; vertex++) {
    System.out.println ("Distance of source " + source + " to "
        + vertex + " vertex is " + D[vertex]);
}
}

public static void main (String [] args) {
    System.out.println ("Roll no: 20115903 ");
    int num_ver = 0, source;
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter the no. of vertices in graph");
    num_ver = sc.nextInt();
    int A [][] = new int [num_ver + 1] [num_ver + 1];
    System.out.println ("Enter the adjacency matrix of the
        graph");
    for (int sn = 1; sn <= num_ver; sn++) {
        for (int dn = 1; dn <= num_ver; dn++) {
            A[sn][dn] = sc.nextInt();
            if (sn == dn) {
                A[sn][dn] = 0;
                continue;
            }
            if (A[sn][dn] == 0) {
                A[sn][dn] = MAX-VALUE;
            }
        }
    }

    System.out.println ("Source vertex: ");
    source = sc.nextInt();
    Experiment18b = new Experiment18b(num_ver);
    b.BellmanfordEvaluation (source, A);
    sc.close();
}

```

```

package lab;
import java.util.*;
import java.util.Scanner;

public class Experiment 19 {
    public static void main (String [] args) {
        System.out.println ("Roll no: 20115403");
        Scanner in = new Scanner (System.in);
        long p, q, d, z, e, n, c;
        int choice;
        System.out.println ("Enter two distinct prime nos");
        p = in.nextLong();
        q = in.nextLong();
        n = p * q;
        z = (p-1) * (q-1);
        System.out.println ("Enter a value of 'd' which is less
            than and relatively prime to " + z + ": ");
        d = in.nextLong();
        for (e = 1; e < z; ++e) {
            if ((e * d) % z == 1) {
                break;
            }
        }
        System.out.println ("p = " + p + "\n q = " + q + "\n n = " + n + "\n z
            = " + z + "\n d = " + d + "\n e = " + e + "\n");
        do {
            System.out.println ("1. Encryption\n 2. Decryption\n
                3. Exit");
            System.out.println ("choose an option (1-3): ");
            choice = in.nextInt();
            switch (choice) {
                case 1: System.out.println ("Enter a plain text");
                    String s = in.next();
                    System.out.println ("Plain Text " + " " + " " + "Cipher
                        text");

```

```

for (int i=0; i<s.length(); i++) {
    long p1 = (int) s.charAt(i);
    c = modexp(p1, e, n);
    System.out.println(s.charAt(i) + "\t\t" + c);
}
break;

```

case 2: System.out.println("Enter a cipher text (0 to stop input):");

```

long [] ci = new long [50];
int j=0;
do {
    ci[j] = in.nextLong();
} while (ci[j++] != 0);
System.out.println("Cipher Text" + "\t" + "Plain Text");
for (int i=0; i<j-1; i++) {
    long p1 = modexp (ci[i], d, n);
    System.out.println(ci[i] + "\t\t" + (char)p1);
}
break;

```

case 3: System.out.println("Program Terminated!!");
System.exit(0);

}

} while (choice != 3);

}

static long modexp(long a, long x, long n){

long r=1;

while (x>0){

if (x%2 == 1){

r = (r * a) % n;

}

a = (a * a) % n;

x /= 2;

}

return (r);

}

}

```

package lab;
import java.io.*;
import java.util.*;
class Queue {
    int q[], j=0, r=0, size;
    void insert (int n){
        Scanner in = new Scanner (System.in);
        q = new int [10];
        for (int i=0; i<n; i++){
            System.out.print ("Enter " + i + " packet: ");
            int ele = in.nextInt();
            if (r+1 > 10){
                System.out.println ("Queue is full! Lost packet: "
                                     + ele);
                break;
            }
            else {
                r++;
                q[r] = ele;
            }
        }
    }
    void delete () {
        Scanner in = new Scanner (System.in);
        Thread t = new Thread();
        if (r == 0){
            System.out.println ("Queue is empty");
        }
        else {
            for (int i=j; i<r; i++){
                try {
                    t.sleep (1000);
                }
            }
        }
    }
}

```



```
catch (Exception e) {}.
```

```
System.out.println (" \n leaked packet: " + q[i]);
```

```
    j++;
```

```
}
```

```
}
```

```
System.out.println ();
```

```
}
```

```
};
```

```
public class Experiment20 extends Thread{
```

```
    public static void main (String [] args) throws Exception{
```

```
        Queue q = new Queue();
```

```
        Scanner sc = new Scanner (System.in);
```

```
        System.out.println ("Roll no.: 20115403");
```

```
        System.out.println ("Enter the total no. of packets  
        to be sent");
```

```
        int size = sc.nextInt();
```

```
        q.insert(size);
```

```
        q.delete();
```

```
    }
```

```
}
```